



Максим Грамин
КРОК

PGDAY'17
RUSSIA

КОНФЕРЕНЦИЯ
ПО БАЗАМ ДАННЫХ

В поисках идеального инструмента



Disclaimer

Личное мнение на личном опыте

Немного о себе

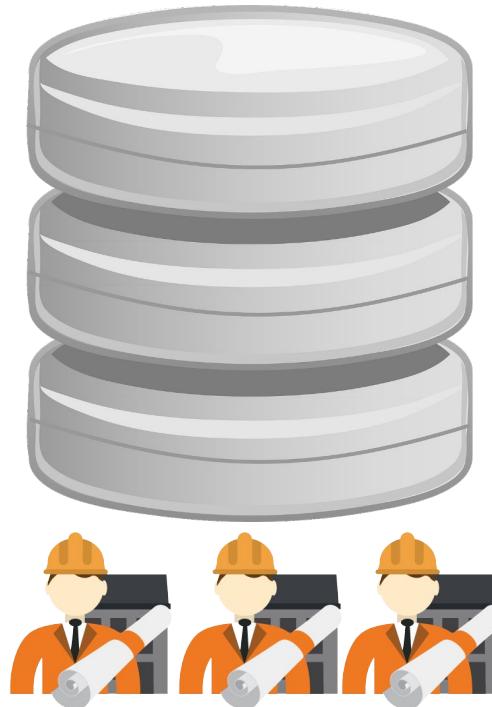
Немного о себе



Немного о себе



Немного о себе



Немного о себе



Однаковые проблемы

- Версионность метаданных
- Версионность данных
- Совместная разработка
- Выбор инструментов

Что будет в докладе

- Анамнез средств разработки БД
- Поиск путей решения
- Анализ одного из вариантов

Чего не будет

- 100%-го готового решения
- «Silver Bullet»
- Рекламы
- Антирекламы

Основные инструменты разработки БД

Version Control

Dump tools

Migration tools

CLI

GUI IDE

Version Control

Indiana Jones and Mutt Williams are shown in a scene from the movie Indiana Jones and the Kingdom of the Crystal Skull. Indiana Jones, played by Harrison Ford, is on the left wearing his signature fedora and trench coat, holding a flashlight. Mutt Williams, played by Shia LaBeouf, is on the right wearing a leather jacket with 'Mutt' written on it. A white speech bubble originates from Indiana Jones's head, containing the text "Кто удалил мой индекс ... ??".

Кто удалил мой
индекс ... ??

Version Control

- ❑ Необходимо использовать

Version Control

- Необходимо использовать
- Сложно

Version Control

- Необходимо использовать
- Сложно
- Все используют по разному

Version Control

- Необходимо использовать
- Сложно
- Все используют по разному
 - только инкремент (миграции)
 - снапшот БД на дату
 - на уровне модели (ORM, db designer etc)

Version Control

- Необходимо использовать
- Сложно
- Все используют по разному
- Где-то вообще не используют



stackoverflow.com: java version control

[Все](#)[Новости](#)[Видео](#)[Картинки](#)[Покупки](#)[Ещё](#)[Настройки](#)[Инструменты](#)

Результатов: примерно 543 000 (0,37 сек.)

version control - Organizing Java projects - Stack Overflow

<https://stackoverflow.com/questions/.../organizing-java-proje...> ▾ Перевести эту страницу

25 дек. 2009 г. - I'm a junior developer and recently started working for a very small ... You seem to be almost exactly in the point where the place I worked at was ...

java - version control for one-man project using ... - Stack Overflow

<https://stackoverflow.com/.../version-control-for-one-man-pr...> ▾ Перевести эту страницу

22 июн. 2010 г. - I'm currently working on several projects on my own (at least the ... It is necessary to

Organizing Java projects



7

I'm a junior developer and recently started working for a very small office where they do a lot of in-house development. I have never worked in a project that involved more than one developer or were as big and complex as these ones.



10

The problem is that they don't use all the tools available (version control, automated building, continuous integration, etc) to their full extent: mainly a project is one big project in eclipse/netbeans using cvs for version control and everything checked in (including library jars), they started using branching for the first time when I started doing branches for small tasks and merging them back. As the projects get bigger and more complex, problems start to arise with dependencies, project structure tied to an IDE, building can be a PITA sometimes, etc. It's hectic at best.

I'm a junior developer and recently started working for a very small office where they do a lot of in-house development. I have never worked in a project that involved more than one developer or were as big and complex as these ones.

The problem is that they don't use all the tools available (version control, automated building, continuous integration, etc) to their full extent: mainly a project is one big project in eclipse/netbeans using cvs for version control and everything checked in (including library jars), they started using branching for the first time when I started doing branches for small tasks and merging them back. As the projects get bigger and more complex, problems start to arise with dependencies, project structure tied to an IDE, building can be a PITA sometimes, etc. It's hectic at best.

[add a comment](#)

10 Answers

active

oldest

votes

Organizing Java projects

I'm a junior developer and recently started working for a very small office where they do a lot of in-

▲
17

You seem to be almost exactly in the point where the place I worked at was when I started there 1,5 years ago, only difference being that you've started toying with branches which is actually something we still don't do at my work but more about that later on in this answer.

▼

Anyway, you're listing a very good set of tools which can help a small company and those work really nicely as subtopics so without further ado,

✓

Version control systems

Most commonly small companies currently use CVS or SVN and there's nothing bad in that, in fact I'd be really worried if no version control was really used at all. However you have to use version control *right*, just having one won't make your life easier. We currently use CVS and are looking into [Mercurial](#), but we've found that the following works as a good set of conventions when working with CVS (and I'd suspect SVN too):

- Have separate users for all committers. It's beyond valuable to know who committed what.
- Don't allow empty commit messages. In fact if possible, configure the repository to reject any commits without comments and/or default comment. *Initial commit for FooBarizer* is better than *Empty log message*
- Use tags to mark milestones, prototypes, alphas, betas, release candidates and final versions. Don't use tags for experimental work or as footnotes/[Post-It notes](#).
- Don't use branches since they really don't work if you're continuing on developing the application. This is mainly because in CVS and SVN branching just doesn't work as expected and it becomes an exercise in futility to maintain any more than two living branches (*head* and *any secondary branch*) over time.

I'm a junior developer or house developer were as big as

The problem continuous in eclipse/netbeans they started up merging them dependencies best.

Google stackoverflow.com: database version control

Все Новости Видео Картинки Покупки Ещё Настройки Инструменты

Результатов: примерно 896 000 (0,50 сек.)

Do you use source control for your database items? - Stack Overflow
<https://stackoverflow.com/.../do-you-use-source-control-for-...> ▾ Перевести эту страницу
Must read Get your database under version control. Check the series of posts by K. Scott Allen. When it comes to version control, the database is often a second ...

Versioning SQL Server database - Stack Overflow
<https://stackoverflow.com/.../versioning-sql-server-database> ▾ Перевести эту страницу

Do you use source control for your database items?



534



297

I feel that my shop has a hole because we don't have a solid process in place for versioning our database schema changes. We do a lot of backups so we're more or less covered, but it's bad practice to rely on your last line of defense in this way.

Surprisingly, this seems to be a common thread. Many shops I have spoken to ignore this issue because their databases don't change often, and they basically just try to be meticulous.

However, I know how that story goes. It's only a matter of time before things line up just wrong and something goes missing.

Are there any best practices for this? What are some strategies that have worked for you?

[database](#)[version-control](#)[share](#) [edit](#) [flag](#)

edited Mar 1 '12 at 21:20

[community wiki](#)

5 revs, 3 users 100%

Brian MacKay

[show 1 more comment](#)[58 Answers](#)

active

oldest

votes

Versioning SQL Server database



267

I want to get my databases under version control. Does anyone have any advice or recommended articles to get me started?



I'll always want to have at least *some* data in there (as [alumb](#) mentions: user types and administrators). I'll also often want a large collection of generated test data for performance measurement.



192

[sql-server](#)[database](#)[svn](#)[version-control](#)[share](#) [edit](#) [flag](#)

edited May 23 at 10:31



Community ♦

1 ● 1

asked Aug 1 '08 at 18:33

[Zack Peterson](#)

27.3k ● 69 ● 183 ● 269

Is there a version control system for database structure changes?

Vote

92

I often run into the following problem.

26

I work on some changes to a project that require new tables or columns in the database. I make the database modifications and continue my work. Usually, I remember to write down the changes so that they can be replicated on the live system. However, I don't always remember what I've changed and I don't always remember to write it down.

19

37

So, I make a push to the live system and get a big, obvious error that there is no `NewColumnX`, ugh.

Regardless of the fact that this may not be the best practice for this situation, is there a version control system for databases? I don't care about the specific database technology. I just want to know if one exists. If it happens to work with MS SQL Server, then great.

sql database oracle version-control

share edit flag

edited Sep 30 '16 at 4:32



Mark Harrison

142k ● 93 ● 241 ● 359

asked Aug 2 '08 at 1:52



EndangeredMassa

12.2k ● 6 ● 38 ● 70

Is there a version control system for database structure changes?

Vote

I often run into the following problem.

Put MySQL database under version control?

Upvote

32

I'm currently using SVN for my PHP projects. I was thinking I should get my database under version control too, but what's the best way to do that? Do I just make a db-folder in my project in SVN, paste SQL changes into a file called from_1.0_to_2.0.sql and commit?

Downvote

[mysql](#)[database](#)[version-control](#)

Share

[edit](#)[flag](#)

6

asked Feb 3 '11 at 8:14



user479911



142k • 93 • 241 • 359



12.2k • 6 • 38 • 70

Is there a version control system for database structure changes?

Oracle: put existing db to version control



I have existing oracle database. I want to put it under source control (Subversion). The only solution I know - create 'DROP/CREATE/INSERT' text scripts and store them to SVN.

5



May be there is better way to manage schema and data? I'm using Oracle SQL Developer and I have seen there Migration/Repository management features. Should I use them? And how to use them?

[oracle](#)[svn](#)[version-control](#)[oracle-sqldesigner](#)[share](#) [edit](#) [flag](#)

asked Aug 14 '12 at 13:27



Igor Konoplyanko

5,382 ● 5 ● 41 ● 77



142k ● 93 ● 241 ● 359



12.2k ● 6 ● 38 ● 70

Проблемы

- Не все используют
- Не все используют правильно
- Не понятно как правильно

DB Dump Tools

Dump tools

- ❑ pg_dump
- ❑ mysqldump
- ❑ dbms_datapump, export/import
- ❑ cqlsh - describe schema

Dump tools



Мощный и крутой инструмент

Dump tools



Мощный и крутой инструмент



Администрирование

Dump tools



Мощный и крутой инструмент



Администрирование



Разработка



Я думал дамп с
прода будет
поменьше !! ...

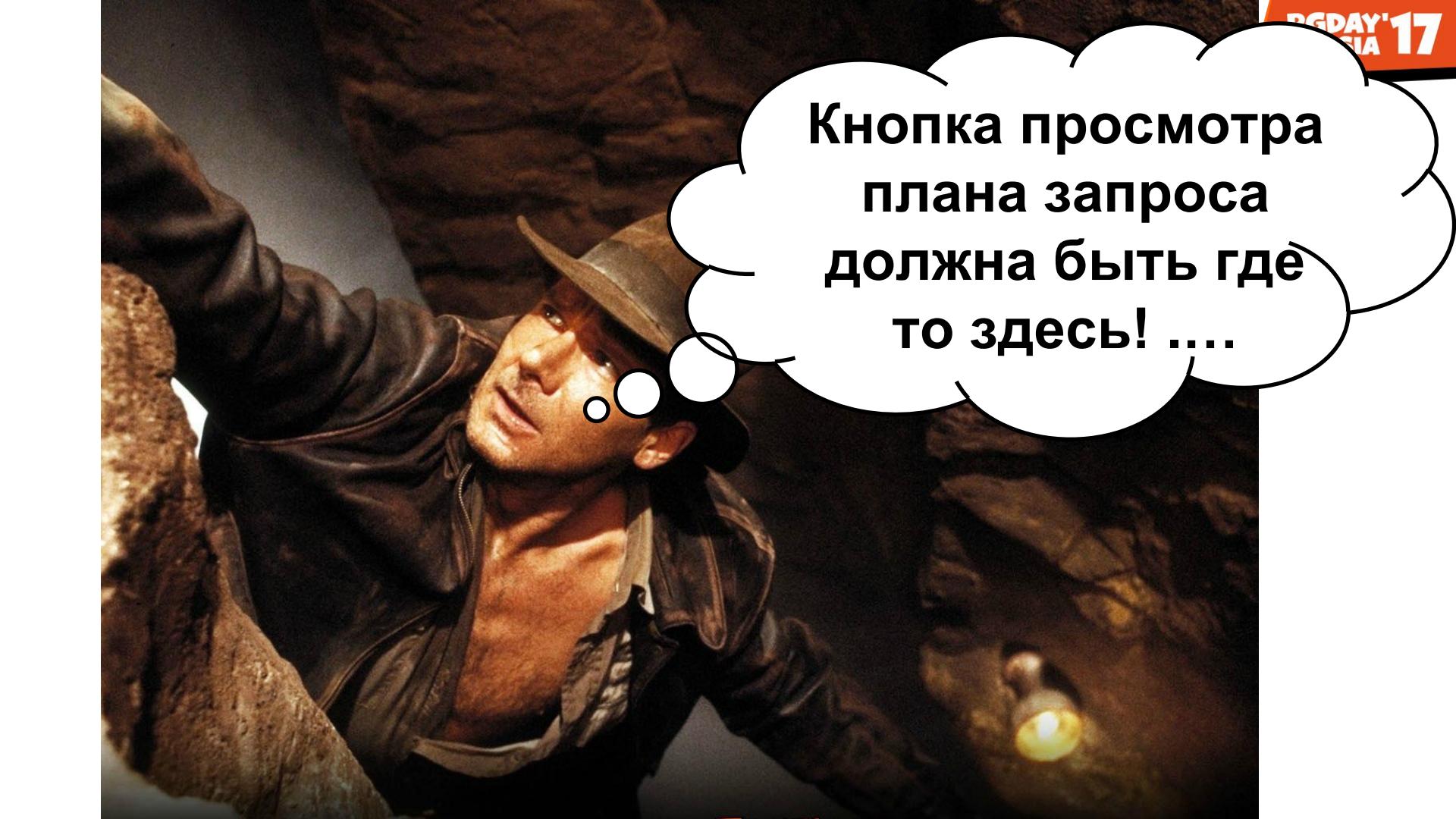
Command Line Utilities

- psql
- sqlplus
- mysql
- cqlsh
- mongo shell

Migration tools

- Liquibase
- Flyway
- Phinx
- Cassandra Migration

GUI IDE



Кнопка просмотра
плана запроса
должна быть где
то здесь!

GUI IDE

NetBeans IDE Dev 20130113-f5bc69ac5215

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Source History Connection: jdbc:derby://localhost:1527/sample [app on APP]

270.1/314.2MB

Projects Files Serv... Favorites

Databases MySQL Server at localhost:3306 [root] (dis... Java DB Drivers jdbc:derby://localhost:1527/sample [app or APP]

APP

- Tables CUSTOMER, DISCOUNT_CODE, MANUFACTURER, MICRO_MARKET, PRODUCT, PRODUCT_CODE, PURCHASE_ORDER
- Views
- Procedures
- NULLID
- SQL
- SYS
- SYSCAT
- SYSCS_DIAG
- SYSCS_UTIL
- SYSFUN
- SYSIBM
- SYSPROC
- SYSTAT

Web Services Servers Maven Repositories Cloud Hudson Builders

Usages Versioning Output

SQL Command 1

```
1 select * from APP.CUSTOMER;
```

CUSTOMER

```
select * from APP.CUSTOMER...
```

DISCOUNT_CODE

MANUFACTURER

MICRO_MARKET

PRODUCT

PRODUCT_CODE

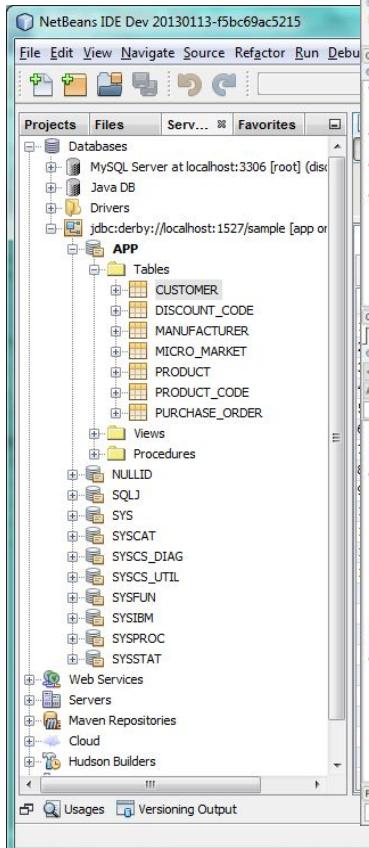
PURCHASE_ORDER

20 Total Rows: 13 Page: 1 of 1 Matching Rows:

#	CUST_ID	DISCOUNT_CODE	ZIP	ADDRESSLINE1	ADDRESSLINE2	CITY
1	1 N	95117		111 E. Las Olivas Blvd	Suite 51	Fort Lauderdale
2	2 M	95035		9754 Main Street	P.O. Box 567	Miami
3	25 M	85638		8989 Red Albatross Drive	Suite 9897	Houston
4	3 L	12347		8585 South Upper Murray Drive	P.O. Box 456	Atlanta
5	36 H	94401		65653 Lake Road	Suite 2323	San Mateo
6	106 L	95035		829 E Flex Drive	Suite 853	San Jose
7	149 L	95117		4381 Kelly Valley Ave	Suite 77	Santa Clara
8	863 N	94401		456 444th Street	Suite 45	Redwood City
9	777 L	48128		West Valley Inc.	Building C	Dearborn
10	753 H	48128		Zed Motor Co	2267 NE Michigan Ave	Dearborn
11	722 N	48124		Big Car Parts	52963 Notouter Dr	Detroit
12	409 L	10095		Old Media Productions	4400 527th Street	Suite 35
13	410 M	10096		Yankee Computer Repair Ltd	9653 211th Ave	Suite 562
					Floor 4	New York

1 | 19 INS

GUI IDE



scott@chicago - PL/SQL Developer Beta

File Project Edit Session Tools Configure View Help Plug-Ins

Clipboard Text Search Selection Navigation Display To-do Items Breakpoints

Connections Chicago scott@chicago sys@chicago Newyork scott@newyork DBA sys@chicago Recent scott@chicago scott@newyork sys@chicago

SQL Output Statistics

```
select empno, mgr, ename, deptno
from emp
where job <> 'MANAGER'
order by ename;
select * from dept
order by deptno;
```

Edit table SCOTT.EMP@CHICAGO

Name	Type	Nullable	Default/Expr.	Generated	On Null	Storage	Comments
EMPNO	NUMBER(4)						The 4-digit number of the employee
ENAME	VARCHAR2(10)	*					The last name of the employee
JOB	VARCHAR2(9)	*	'CLERK'				The job of the employee
MGR	NUMBER(4)	*					The employee number of the manager

employee.pck

employee

Declaration Function Code section Select

DeptName(p_empno in emp.empno%type) return dept.dname%type is

```
-- Return the name of the department
function DeptName(p_empno in emp.empno%type) return dept.dname%type is
begin
  if p_empno is null then
    -- If empno is null, return an unknown name
    result := 'Unknown';
  else
    -- Fetch the name of the department
    select dept.dname
    into result
    from dept,
        emp
    where emp.empno = p_empno
      and dept.deptno = emp.deptno;
  end if;
  return(result);
end;
```

Exception handler

MgrName

FetchRecord

InsertRecord

UpdateRecord

DeleteRecord

LockRecord

empno number(4)
ename varchar2(10)
job varchar2(9)
mgr number(4)
hiredate date

10 PL/SQL component DEPT must be declared
20 PL/SQL:ORA-09064: 'EMP': invalid identifier

20:32 scott@chicago Compiled with errors

1 | 19 INS

GUI IDE

The screenshot displays two side-by-side software environments for database development:

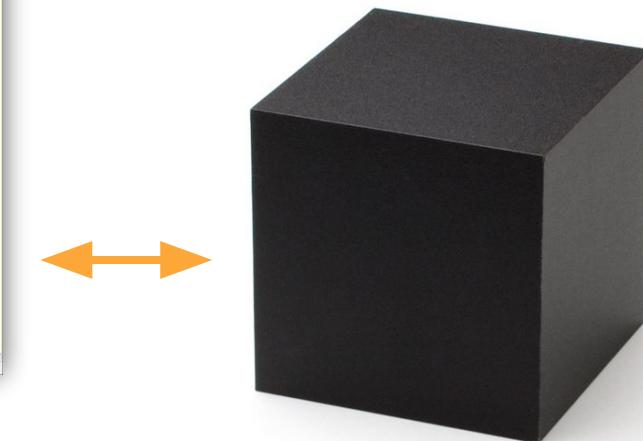
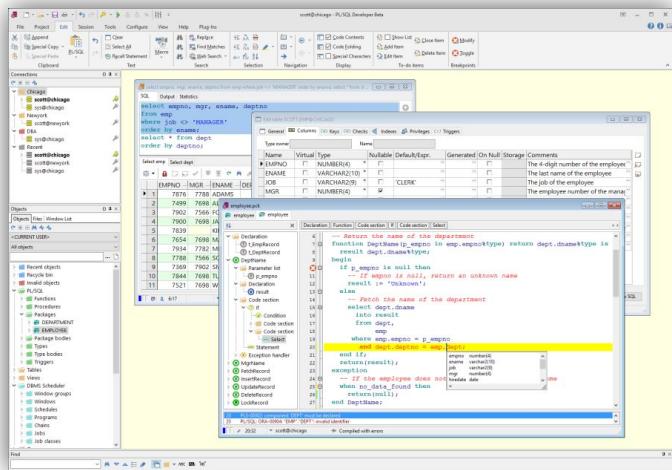
- Toad for Oracle - [QUEST_PERF@DEMO - Editor (FUNC_SAMPLE_7)]**: A window titled "Toad for Oracle" showing the PL/SQL code for a function named FUNC_SAMPLE_7. The code uses cursor c1 to query EMPLOYEE, GRADE, and DEPARTMENT tables based on input parameters A and B.
- NetBeans IDE**: A window titled "NetBeans IDE" showing the same PL/SQL code in a text editor. The interface includes a toolbar, menu bar, and various tool windows like Navigator, Watches, and Output.

A vertical yellow sidebar on the right contains a forum titled "Forums" with several posts:

- Integration between Toad and TortoiseSVN
- Date format problem when exporting to Excel instance
- Assign a color to TOAD when starting from shortcut with connection-info
- Toad 11 autocommit

At the bottom of the screen, there is a status bar with the text "1 | 19 | INS".

- pgAdmin DevCenter dbForge
- MySQL Workbench Robomongo Navicat
- phpMyAdmin Neo4j browser Toad
- IBExpert HareDB HeidiSQL
- SQL Developer Dbeaver
- PLSQL Developer

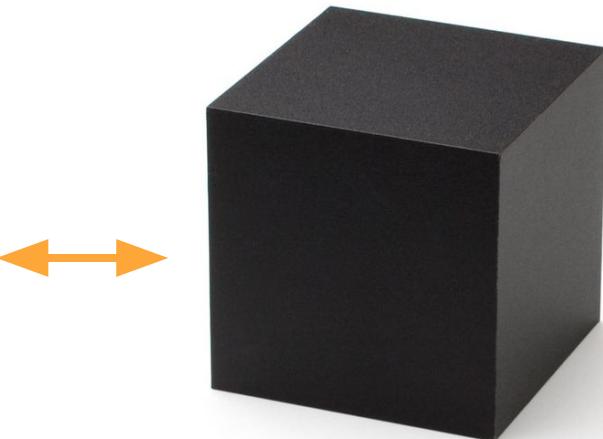
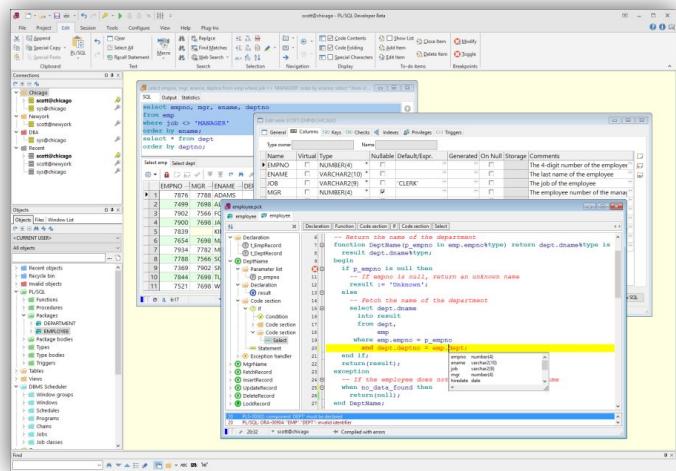


SQL, *QL,
JDBC, ODBC,
ADO.NET, OCI,
etc

GUI IDE



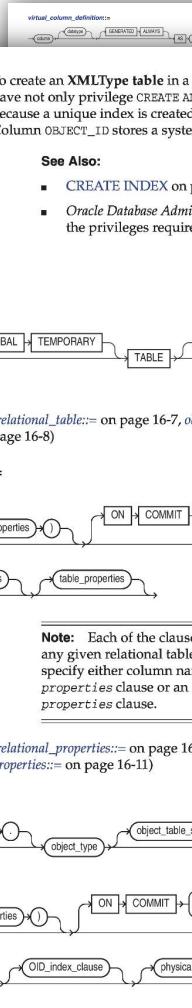
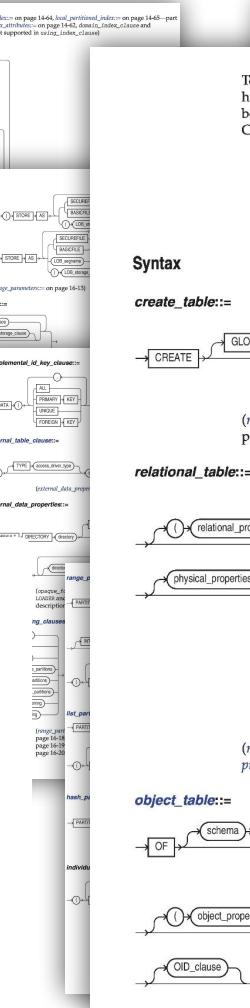
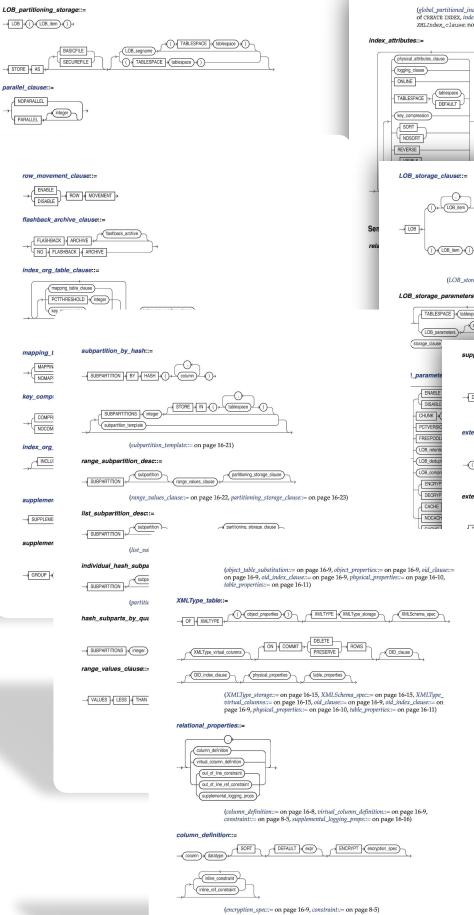
SQL
(DML, DDL, DCL, TCL)



SQL, *QL,
JDBC, ODBC,
ADO.NET, OCI,
etc



GUI IDE



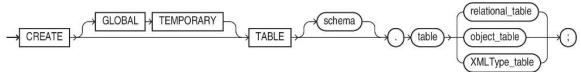
To create an **XMLType** table in a different database schema from your own, you must have not only privilege `CREATE ANY TABLE` but also privilege `CREATE ANY INDEX`. This is because a unique index is created on column `OBJECT_ID` when you create the table. Column `OBJECT_ID` stores a system-generated object identifier.

See Also:

- CREATE INDEX** on page 14-60
Oracle Database Administrator's Guide for more information about the privileges required to create tables using types

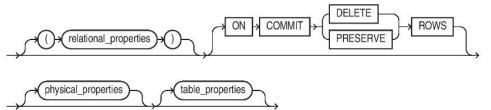
Syntax

create table::=



(*relational_table*::= on page 16-7, *object_table*::= on page 16-7, *XMLType_table*::= on page 16-8)

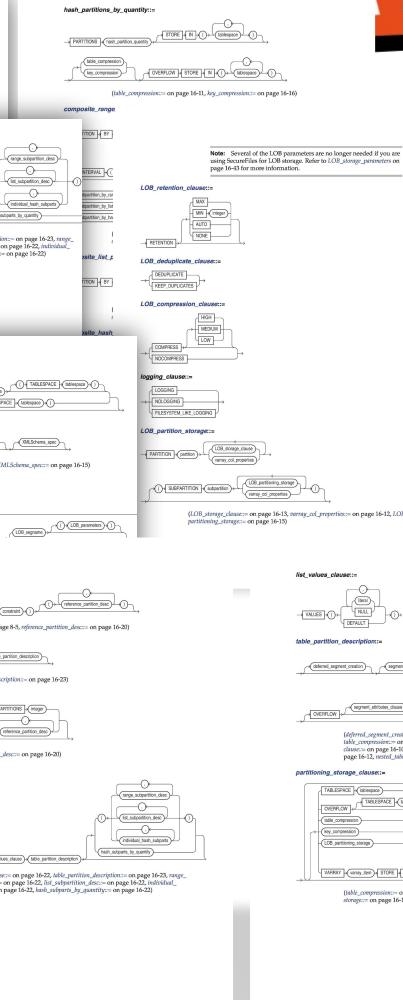
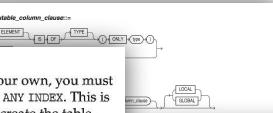
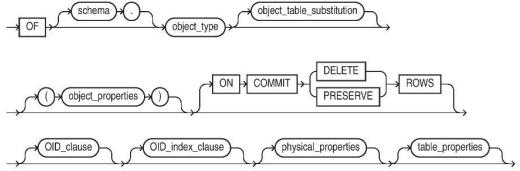
relational table::=



Note: Each of the clauses following the table name is optional for any given relational table. However, for every table you must at least specify either column names and data types using the *relational_properties* clause or an AS *subquery* clause using the *table_properties* clause.

(*relational_properties*::= on page 16-8, *physical_properties*::= on page 16-10, *table_properties*::= on page 16-11)

object table::=



Проблемы

- ❑ «All inclusive»

Проблемы

- ❑ «All inclusive»
- ❑ Вторичность кода

Проблемы

- «All inclusive»
- Вторичность кода
- Платные/закрытые

Проблемы

- ❑ «All inclusive»
- ❑ Вторичность кода
- ❑ Платные/закрытые
- ❑ Расширяемость

Проблемы

- «All inclusive» Интеграция
- Вторичность кода
- Платные/закрытые
- Расширяемость

Проблемы

- «All inclusive»
- Вторичность кода
- Платные/закрытые
- Расширяемость
- Интеграция
- Платформозависимость

Проблемы

- «All inclusive» Интеграция
- Вторичность кода Платформозависимость
- Платные/закрытые Много
- Расширяемость

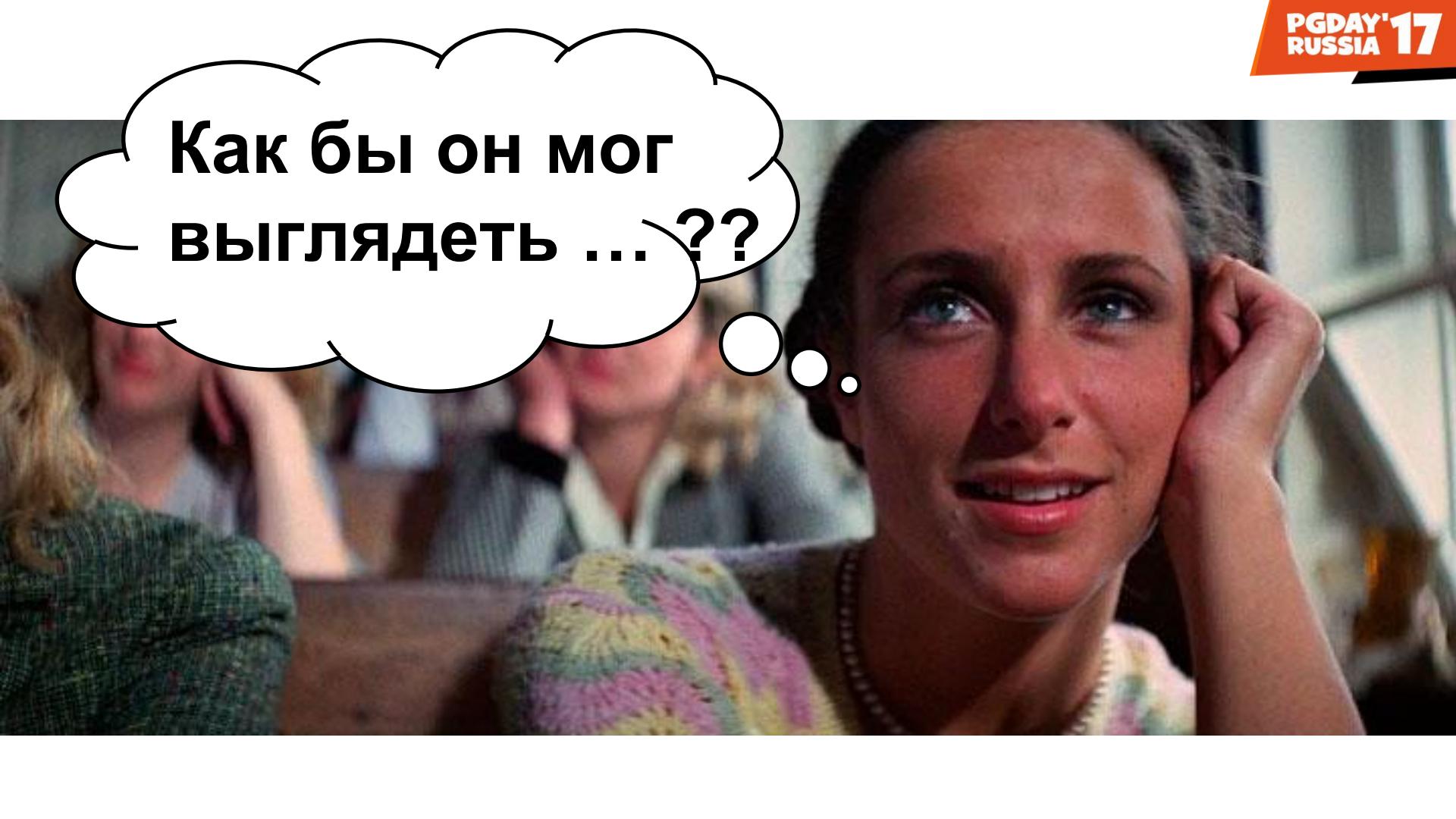
Итого

- «Polyglot Persistence»
- «Лихие 90-е»
- «Импортозамещение»



A close-up of Harrison Ford as Rick Blaine from the movie Casablanca. He is wearing a fedora and a light-colored trench coat. A speech bubble originates from his mouth, containing the text in Russian.

«... критикуешь –
предлагай ...»



Как бы он мог
выглядеть ... ??

Как бы он мог выглядеть ... ??

- Полностью конфигурируемый
- Прозрачный (transparent)
- Любой flow (code first, db first, model first)

«EVERYTHING AS A CODE»

«Everything as a Code»

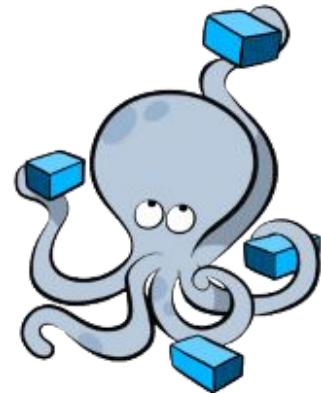
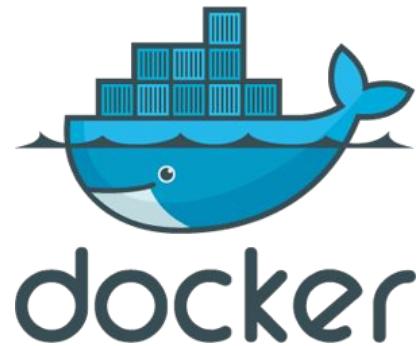
«Everything as a Code»



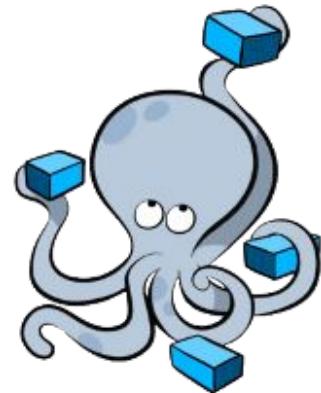
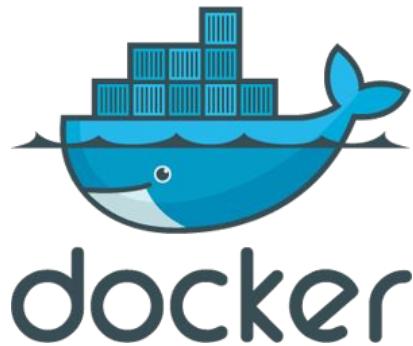
«Everything as a Code»



«Everything as a Code»



«Everything as a Code»



MESOS





И причем тут
базы данных ?

«Everything as a Code»

«Everything as a Code»

- ❑ Исходный код объектов БД

«Everything as a Code»

- ❑ Исходный код объектов БД
- ❑ Код манипуляции объектами БД

«Everything as a Code»

- ❑ Исходный код объектов БД
- ❑ Код манипуляции объектами БД
- ❑ Структура

«Everything as a Code»

- Исходный код объектов БД
- Код манипуляции объектами БД
- Структура
- Шаблоны

«Everything as a Code». SQL. Postgresql

```
select t.table_schema as "schema"
      , t.table_name      as "table"
  from information_schema.tables t
 where t.table_type = 'BASE TABLE'
   and t.table_schema like '$schema'
   and t.table_name  like '$table'
```

«Everything as a Code». SQL. Oracle

```
select t.owner           as "schema"
      , t.table_name    as "table"
  from all_tables t
 where t.owner like '$schema'
   and t.table_name like '$table'
```

«Everything as a Code». SQL. Cassandra

```
select keyspace_name      as "schema"
      , columnfamily_name as "table"
from system.schema_columnfamilies
where keyspace_name = '$schema'
  and columnfamily_name = '$table'
```

«Everything as a Code». SQL

```
alter table persons add column (age number);
```

```
alter table persons disable all triggers;
```

```
drop table persons;
```

```
exec dbms_stats.gather_stats('hr', 'persons');
```

«Everything as a Code». SQL

```
alter database open read only;
```

```
alter system flush buffer_cache;
```

```
alter tablespace users offline;
```

```
alter session set sql_trace = true;
```

«Everything as a Code». Структура

```
<type name="SCHEMA">
    <type name="TABLE">
        <type name="COLUMN" />
        <type name="PK" />
        <type name="INDEX" />
    </type>
    <type name="VIEW" />
    <type name="PROCEDURE" />
</type>
```

«Everything as a Code». Структура

```
<type name="SCHEMA">
    <type name="TABLE">
        <type name="COLUMN" />
        <type name="PK" />
        <type name="INDEX" />
    </type>
    <type name="VIEW" />
    <type name="PROCEDURE" />
</type>
```

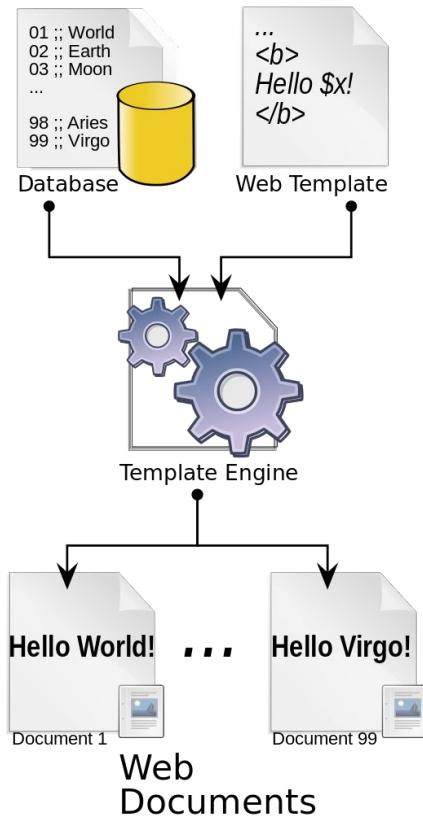
«Everything as a Code». Структура

```
<type name="SCHEMA">
    <type name="TABLE">
        <type name="COLUMN" />
        <type name="PK" />
        <type name="INDEX" />
    </type>
    <type name="VIEW" />
    <type name="PROCEDURE" />
</type>
```

«Everything as a Code». Files

```
<database_1>
  |__ <schema_1>
    |  |__ <table_1>
    |    |__ <ddl.sql>
    |    |__ <pk.sql>
    |    |__ <indexes.sql>
    |    |__ <data.sql>
    |    |__ <test_data.sql>
    |    |__ <views>
    |      |__ <table_1_vw.sql>
    |    |__ <procedures>
    |      |__ <get_all_data_prc.sql>
    ...
  ...
```

«Everything as a Code». Шаблоны

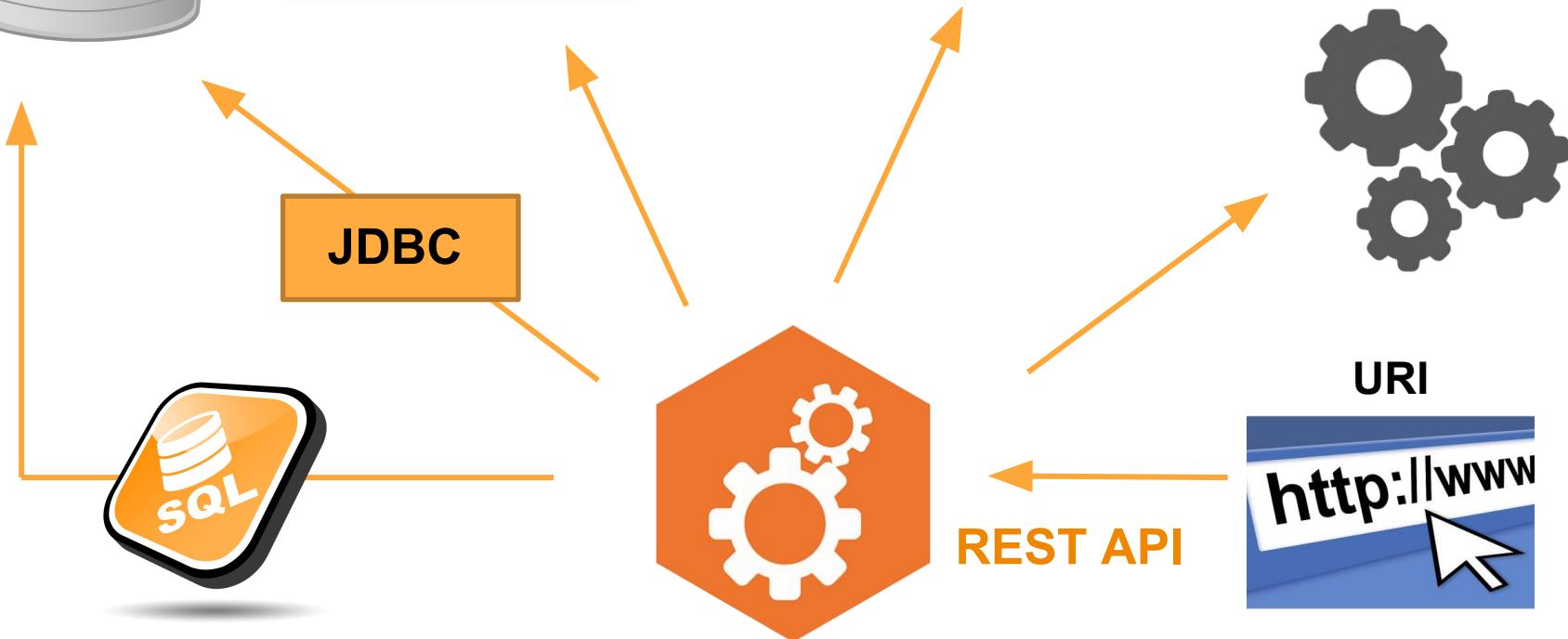
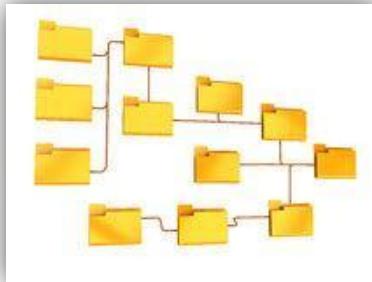


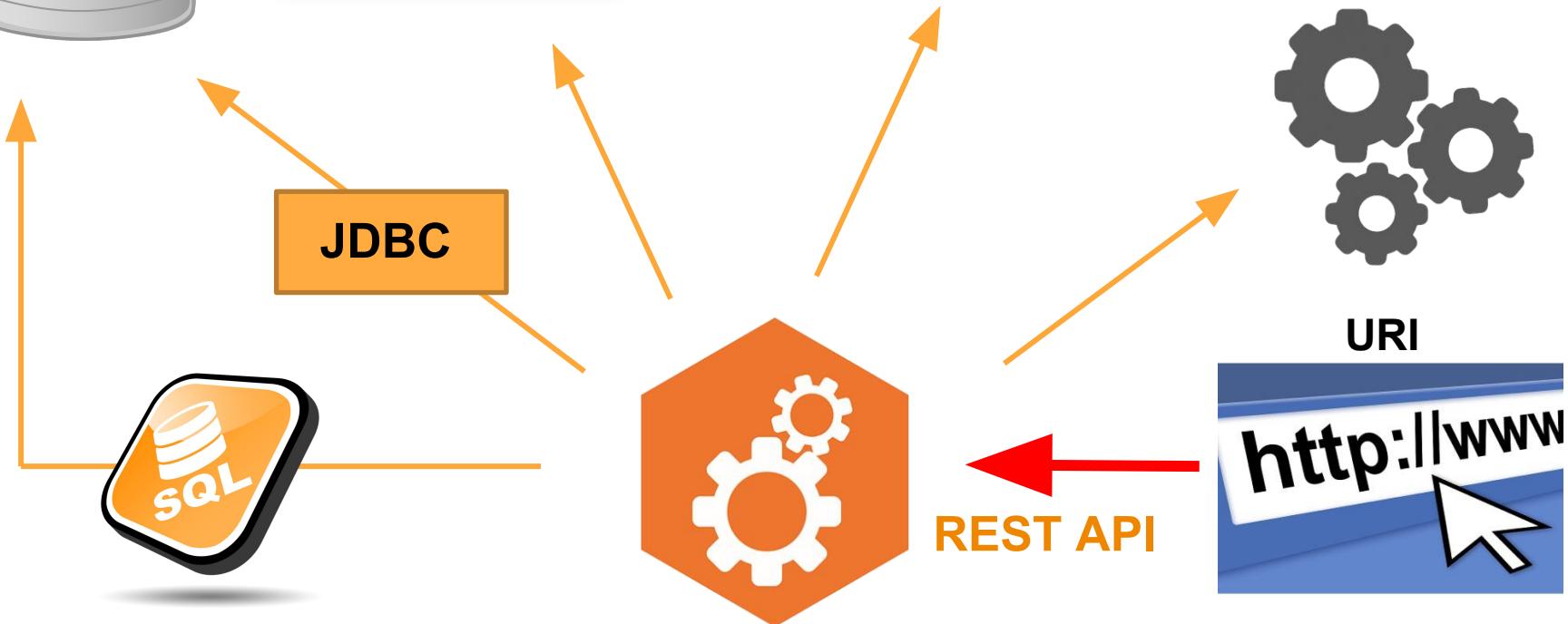
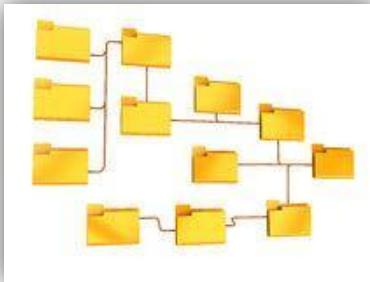
«Everything as a Code». Rest

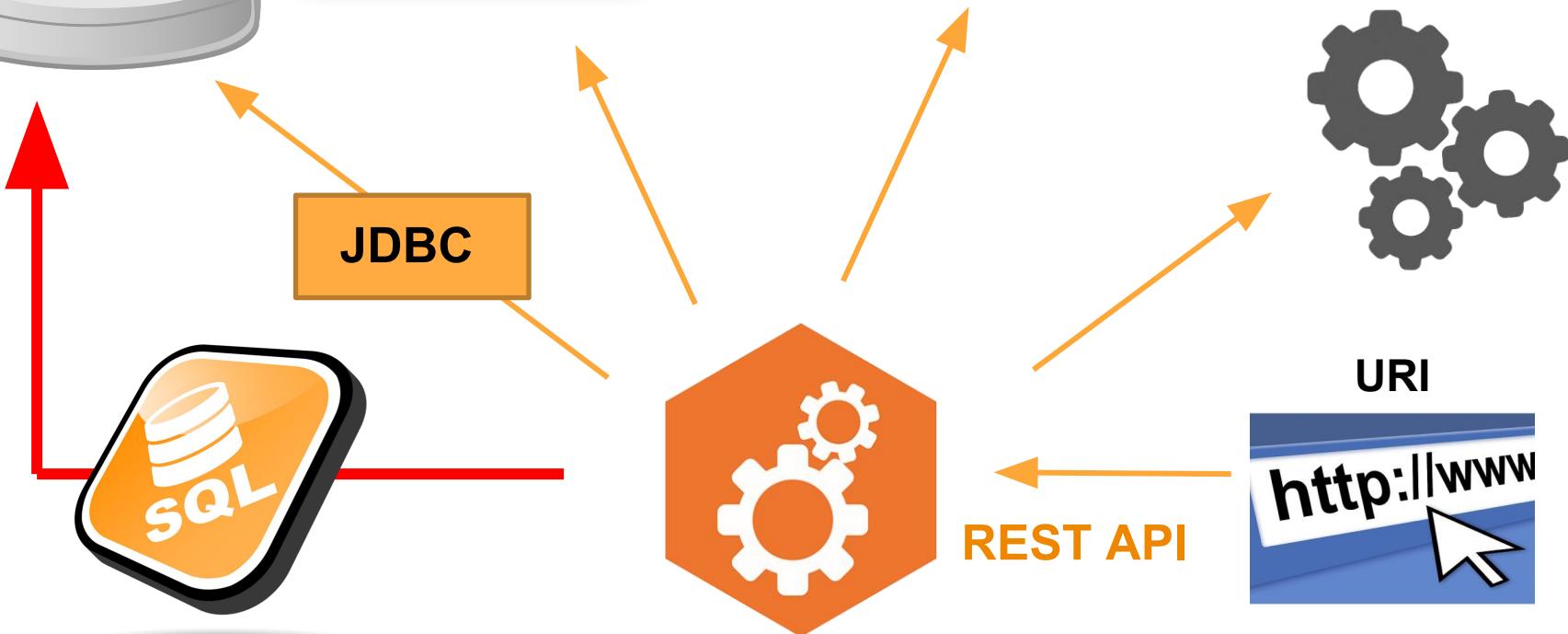
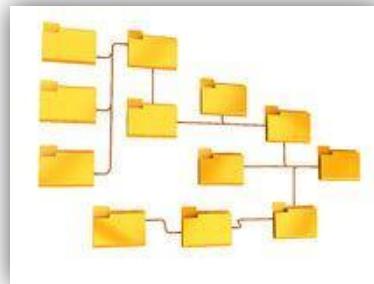


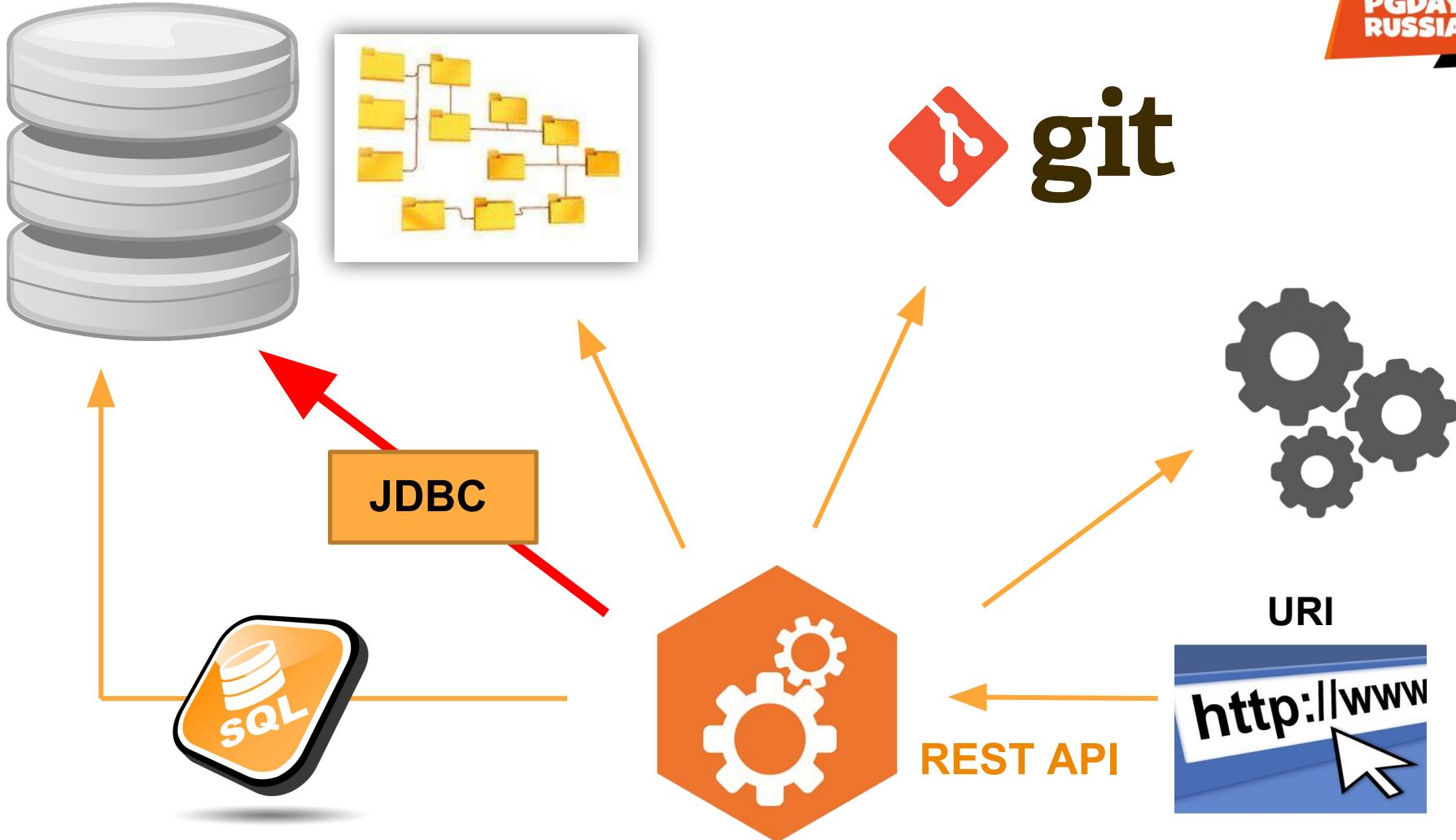
«Everything as a Code». Rest

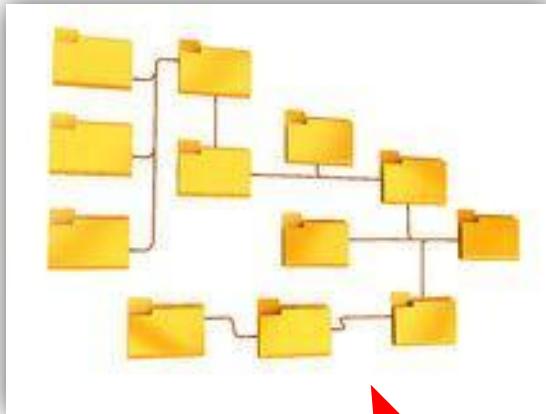
- Объекты БД как ресурсы
- Человекопонятный
- Простая интеграция
- Расширяемый
- Прозрачное кеширование



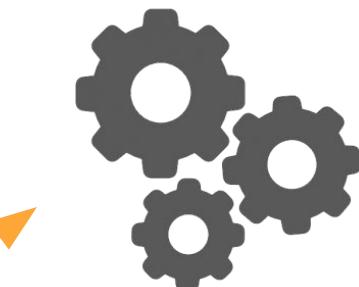






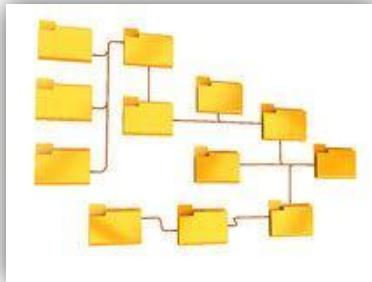


REST API

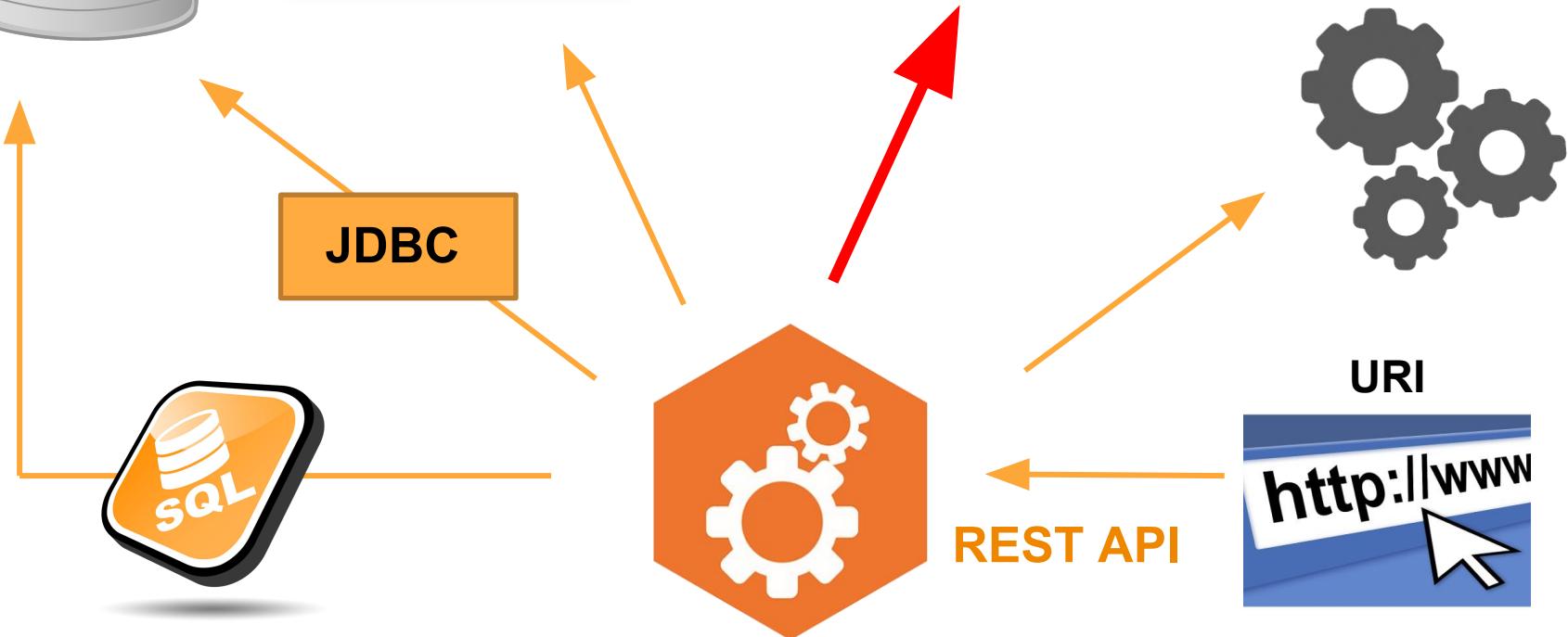


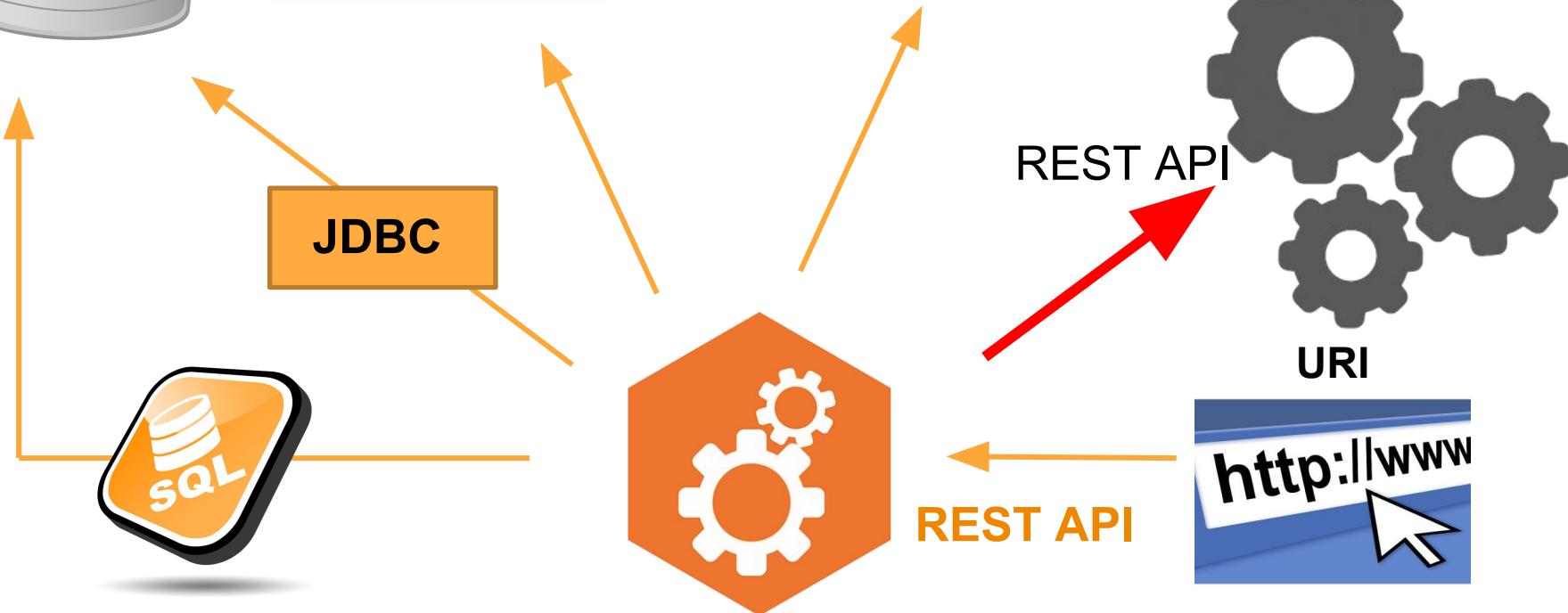
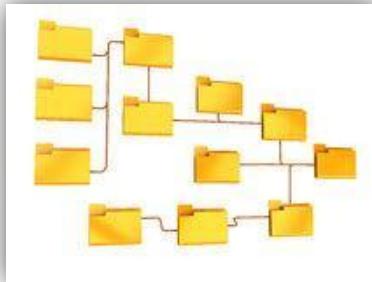
URI

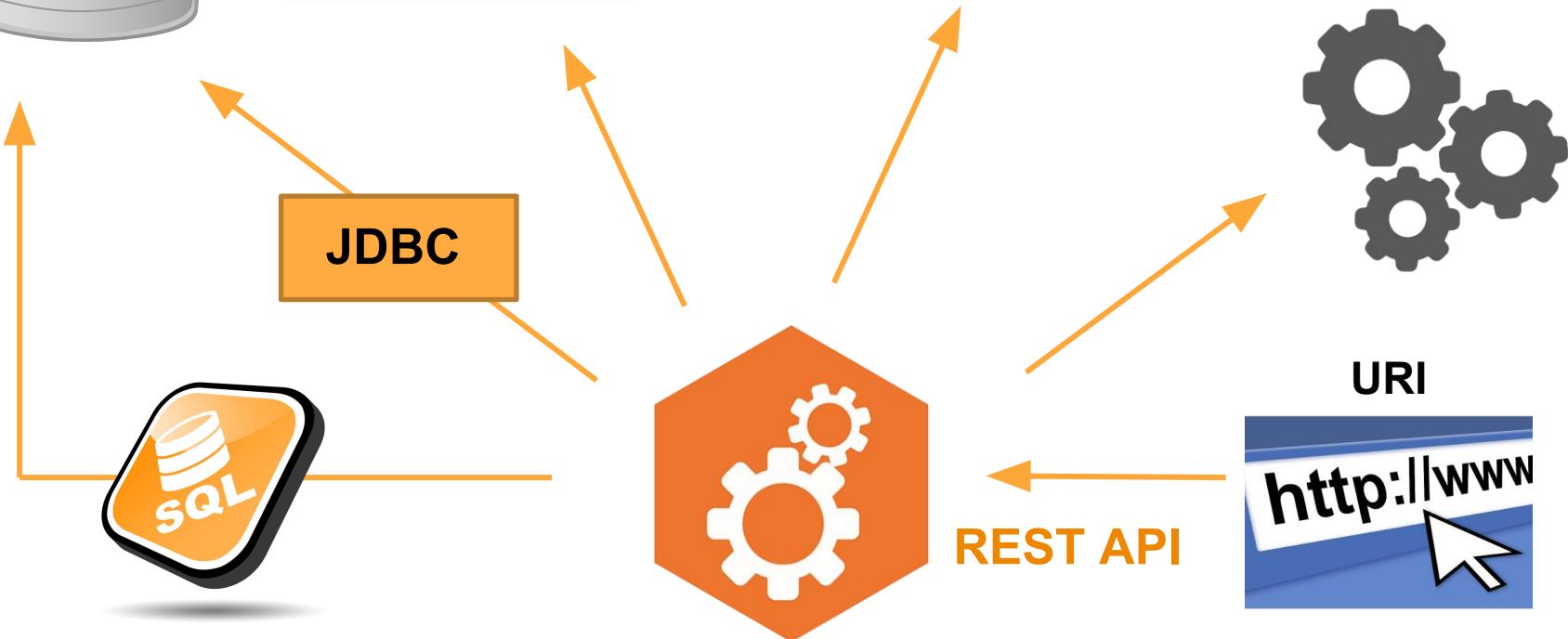
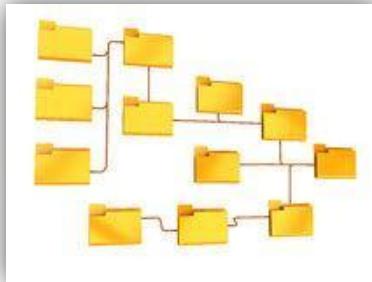




git







URI schema

URI schema

type/resource_path method

URI schema

type/resource_path method

URI schema

type/resource_path method
schema

URI schema

type/resource_path method

schema

table

URI schema

type/resource_path method

schema

table

idx

URI schema

type/resource_path method

schema

table

idx

etc

URI schema

type/resource_path method

schema

table

idx

etc

URI schema

type/resource_path method

schema /hr

table

idx

etc

URI schema

type/resource_path method

schema /hr

table /hr.persons

idx

etc

URI schema

type/resource_path method

schema	/hr
table	/hr.persons
idx	/hr.persons.prs_idx
etc	

URI schema

type/resource_path method

schema	/hr
table	/hr.persons
idx	/hr.persons.prs_idx
etc	etc

URI schema

type/resource_path method

schema	/hr
table	/hr.persons
idx	/hr.persons.prs_idx
etc	etc

URI schema

	type/resource_path	method
schema	/hr	-X GET
table	/hr.persons	
idx	/hr.persons.prs_idx	
etc	etc	

URI schema

	type/resource_path	method
schema	/hr	-X GET
table	/hr.persons	-X POST
idx	/hr.persons.prs_idx	
etc	etc	

URI schema

	type/resource_path	method
schema	/hr	-X GET
table	/hr.persons	-X POST
idx	/hr.persons.prs_idx	-X DELETE
etc	etc	

URI schema

	type/resource_path	method
schema	/hr	-X GET
table	/hr.persons	-X POST
idx	/hr.persons.prs_idx	-X DELETE
etc	etc	etc

```
curl -X GET api/index/bookings.tickets*.*_idx
```

```
select i.schemaname      as "schema"
      , i.tablename     as "table"
      , i.indexname    as "index"
  from pg_catalog.pg_indexes i
 where i.schemaname like 'bookings'
   and i.tablename   like 'tickets%'
   and i.indexname  like '%_idx'
```

```
curl -X GET api/index/bookings.tickets*.*_idx
```

```
select i.schemaname      as "schema"  
      , i.tablename       as "table"  
      , i.indexname       as "index"  
  from pg_catalog.pg_indexes i  
 where i.schemaname like 'bookings'  
   and i.tablename    like 'tickets%'  
   and i.indexname    like '%_idx'
```

```
curl -X GET api/index/bookings.tickets*.*_idx
```

```
select i.schemaname      as "schema"
      , i.tablename       as "table"
      , i.indexname       as "index"
  from pg_catalog.pg_indexes i
 where i.schemaname      like 'bookings'
   and i.tablename       like 'tickets%'
   and i.indexname       like '%_idx'
```

```
curl -X GET api/index/bookings.tickets*.*_idx
```

```
select i.schemaname      as "schema"
      , i.tablename       as "table"
      , i.indexname       as "index"
  from pg_catalog.pg_indexes i
 where i.schemaname
   and i.tablename
   and i.indexname
     like 'bookings%'
     like 'tickets%'
     like '%_idx'
```

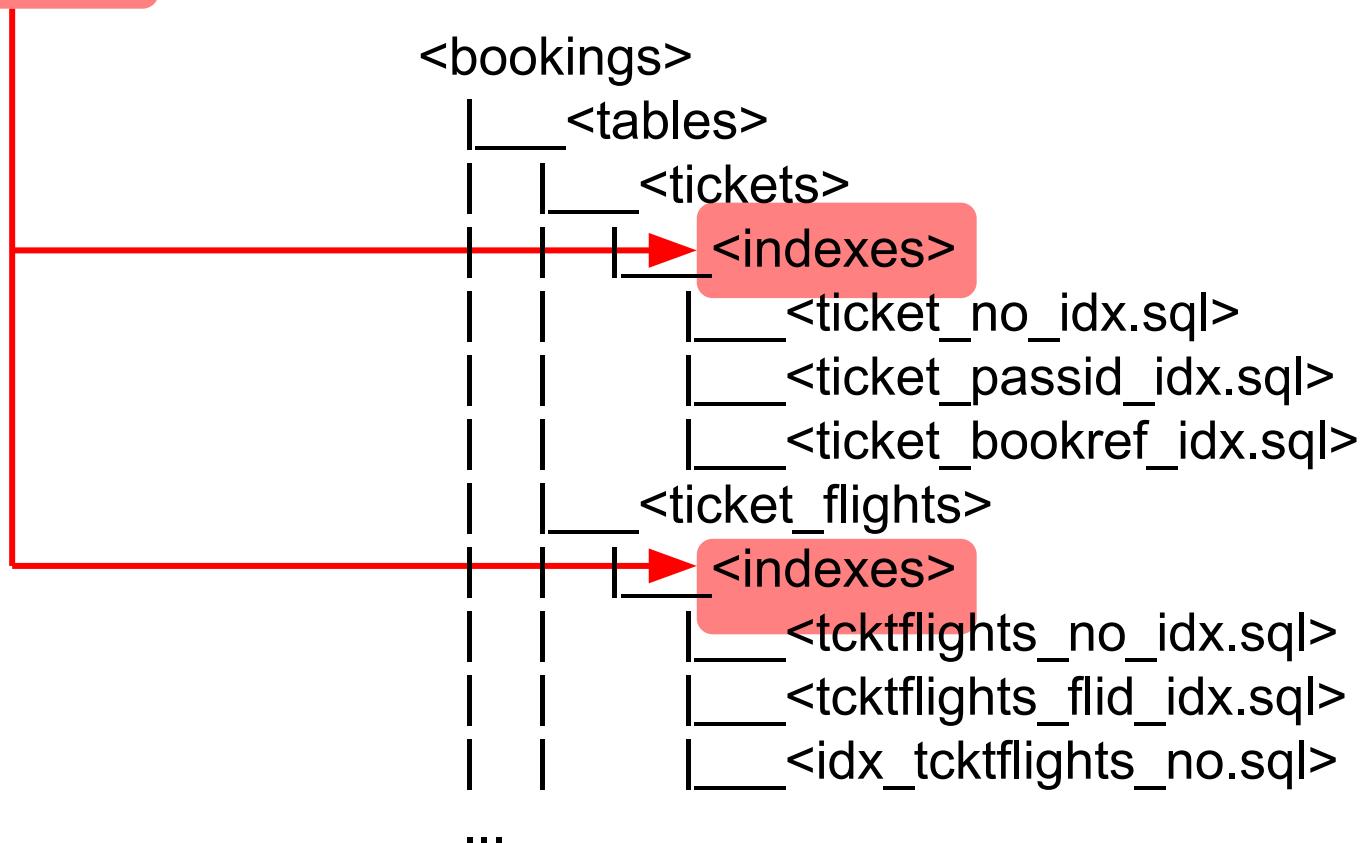
```
curl -X GET api/index/bookings.tickets*.*_idx
```

```
select i.schemaname      as "schema"
      , i.tablename       as "table"
      , i.indexname       as "index"
  from pg_catalog.pg_indexes i
 where i.schemaname like 'bookings'
   and i.tablename    like 'tickets%'
   and i.indexname    like '%_idx'
```

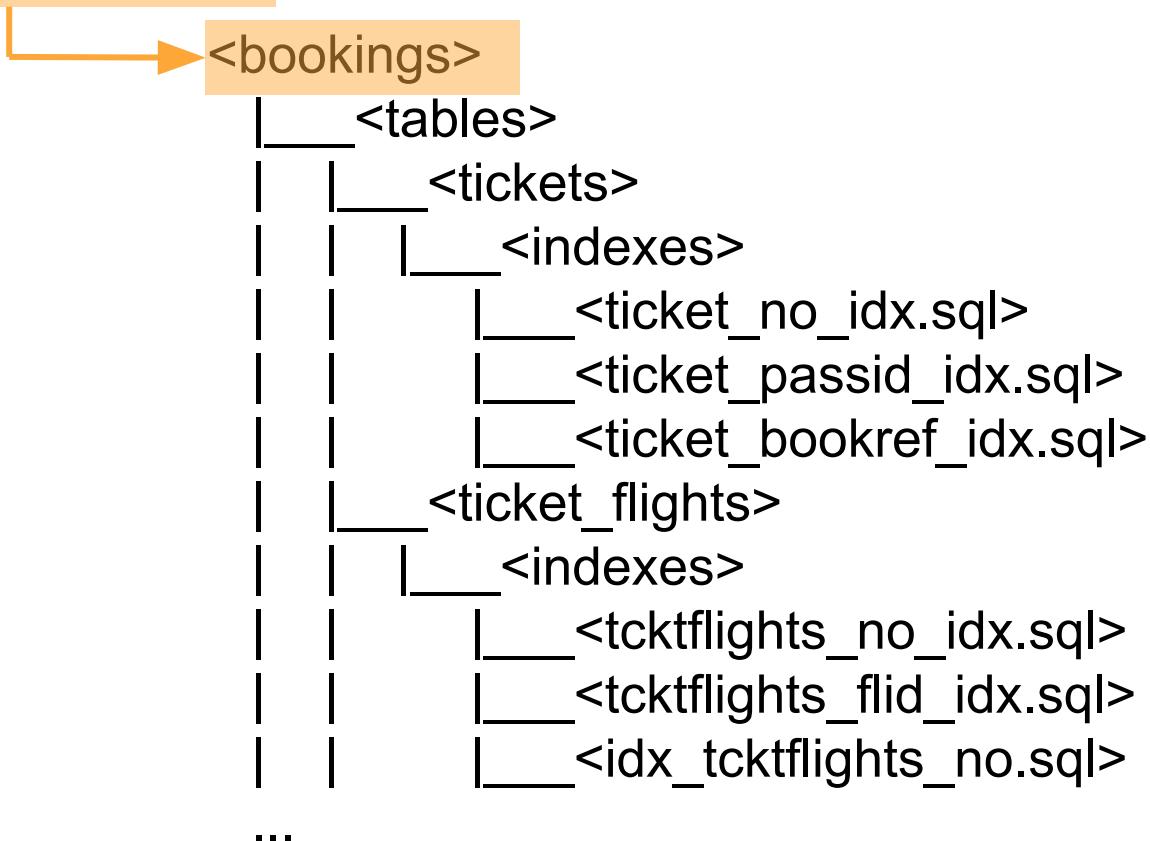
GET index/bookings.tickets*.*_idx

```
<bookings>
  |__ <tables>
    |__ <tickets>
      |__ <indexes>
        |__ <ticket_no_idx.sql>
        |__ <ticket_passid_idx.sql>
        |__ <ticket_bookref_idx.sql>
      |__ <ticket_flights>
        |__ <indexes>
          |__ <tcktflights_no_idx.sql>
          |__ <tcktflights_flid_idx.sql>
          |__ <idx_tcktflights_no.sql>
...
...
```

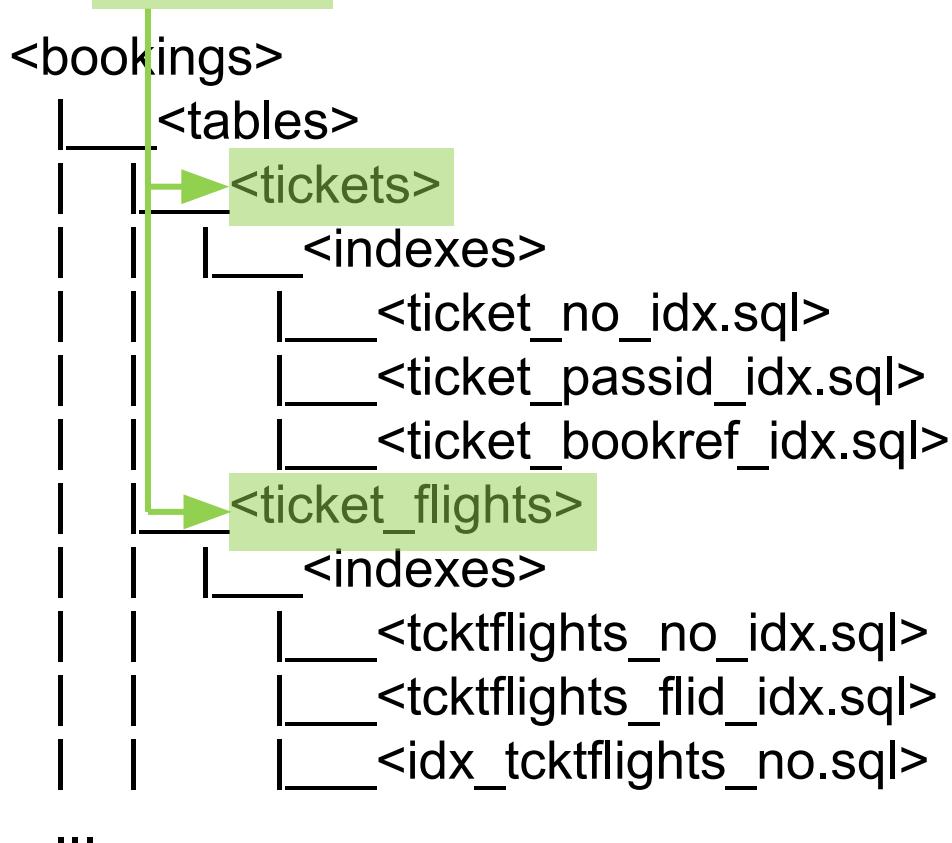
GET index/bookings.tickets*.*_idx



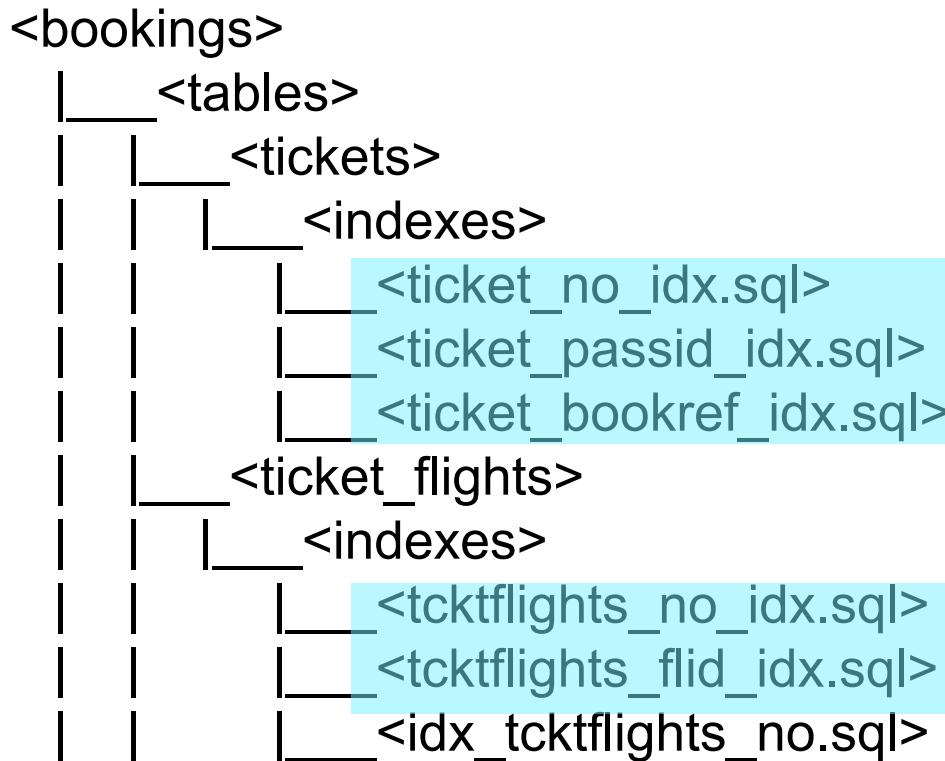
GET index/bookings.tickets*.*_idx



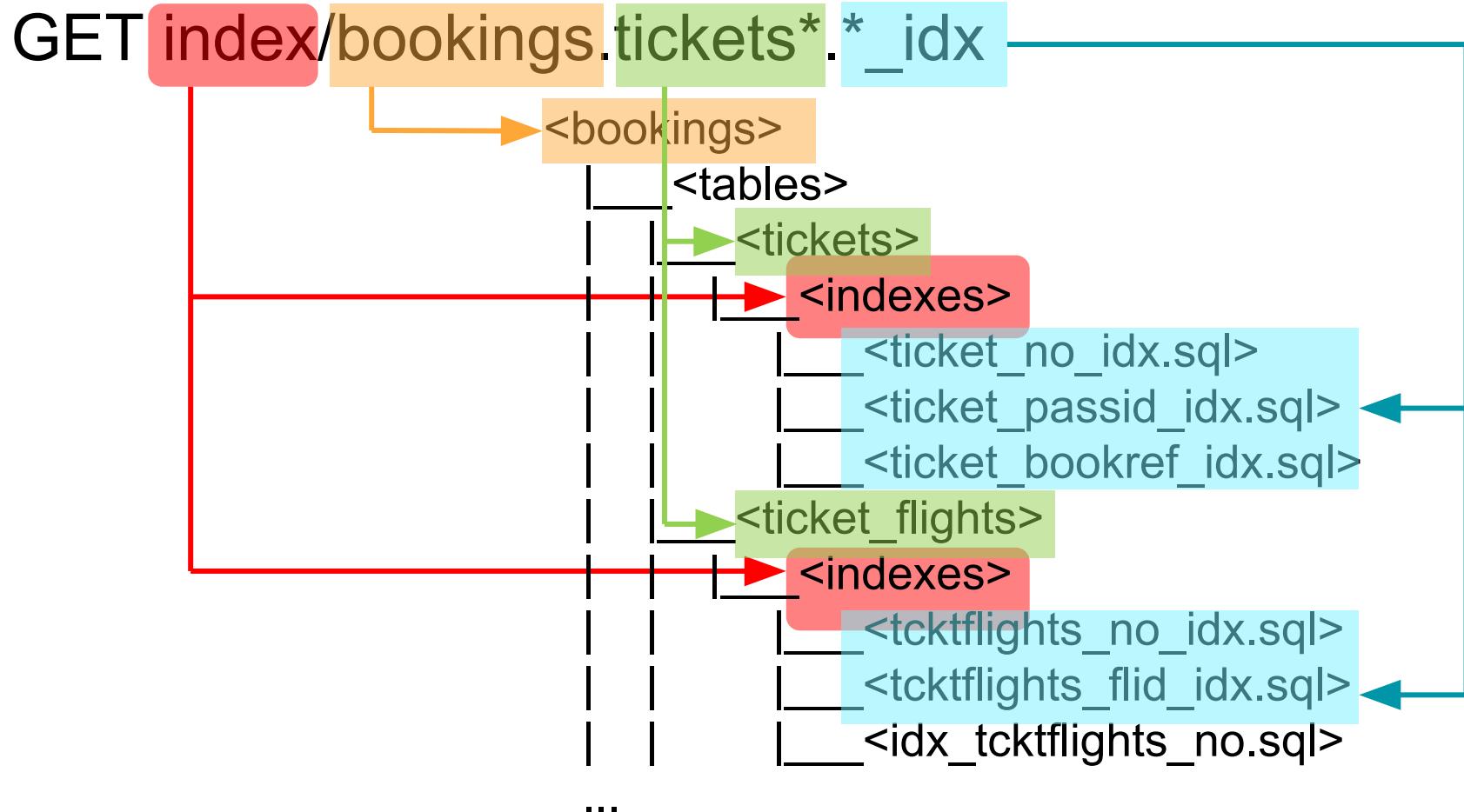
GET index/bookings.tickets*.*_idx



GET index/bookings.tickets*.*_idx



...



Шаблоны

Шаблон

```
CREATE TABLE "${table.schema}"."${table.name}" (
<#list table.columns as column>
    "${column.name}" ${column.type} ${column.size},
</#list>
) ;
```

```
CREATE TABLE "HR"."EMPLOYEES" (
    "EMPLOYEE_ID"      NUMBER(6,0),
    "FIRST_NAME"        VARCHAR2(20),
    "LAST_NAME"         VARCHAR2(25),
    "EMAIL"             VARCHAR2(25),
    "PHONE_NUMBER"      VARCHAR2(20),
    "HIRE_DATE"         DATE,
    "JOB_ID"            VARCHAR2(10),
    "SALARY"             NUMBER(8,2));
```

curl -X GET table/hr.*/relations

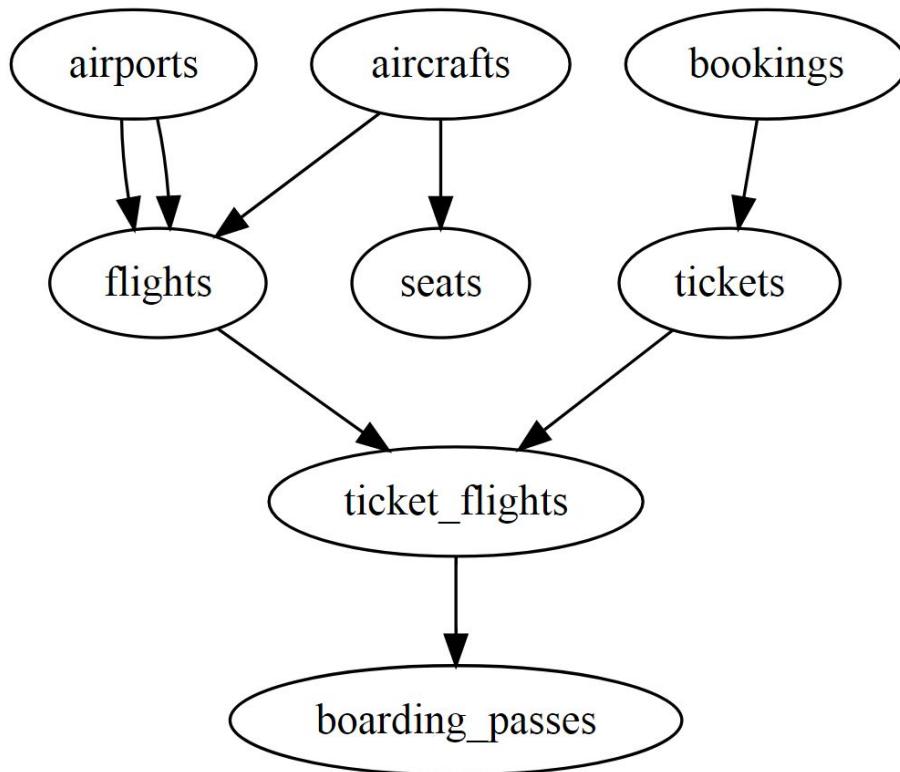
```
select s.relname as master_table
      , t.relname as slave_table
  from pg_catalog.pg_constraint c
  join pg_catalog.pg_class s on s.oid = c.confrelid
  join pg_catalog.pg_class t on t.oid = c.conrelid
 where c.contype = 'f' and s.relkind = 'r'
```

```
digraph DbObjectsRelations {
  <#list tables as table>
    "${table.master_table}" -> "${table.slave_table}"
  </list>
}
```

```
curl -X GET table/hr.*/relations
```

```
digraph DbObjectsRelations {  
    "aircrafts"      -> "flights"  
    "airports"        -> "flights"  
    "airports"        -> "flights"  
    "aircrafts"       -> "seats"  
    "flights"         -> "ticket_flights"  
    "tickets"         -> "ticket_flights"  
    "bookings"        -> "tickets"  
    "ticket_flights"  -> "boarding_passes"  
}
```

```
curl -X POST -H "Accept: image/png" table/hr.*/relations
```





CHEF



ANSIBLE



ATOM YEOMAN JFrog Artifactory





«... предлагаешь
– делай ...»

Ядро

[Code](#)[Issues 16](#)[Pull requests 0](#)[Projects 0](#)[Wiki](#)[Settings](#)[Insights](#)

DB-tool for Developers, DBA and DevOps

[Edit](#)[schema](#) [devops](#) [java](#) [dba](#) [sql](#) [ddl](#) [db](#) [nosql](#) [rdbms](#) [db-manager](#) [db-migration](#) [db-generator](#) [sql-generation](#) [db-swiss-knife](#)
[db-ide](#) [db-tool](#) [code-generator](#) [Manage topics](#)[303 commits](#)[3 branches](#)[1 release](#)[2 contributors](#)[MIT](#)

Branch: master ▾

[New pull request](#)[Create new file](#)[Upload files](#)[Find file](#)[Clone or download](#) ▾

 mgramin	update Readme	Latest commit 2a46a82 11 hours ago
 src	fix #62	11 hours ago
 .gitattributes	add .gitattributes file	a month ago
 .gitignore	move lib folder to assembly static files	14 days ago
 .travis.yml	update docker file	14 days ago
 LICENSE	fix some checkstyle warnings	a month ago
 README.md	update Readme	11 hours ago
 pom.xml	fix #53	14 days ago

Конфигурации



sql-boot

Repositories People 1 Teams 0 Projects 0 Settings

Search repositories... Type: All ▾

oracle
Updated 3 days ago

cassandra
Updated 3 days ago

postgres
Updated 3 days ago

Project

postgres

.git

conf

application-postgres.yml

config.xml

table.xml

.travis.yml

docker-compose.yml

Dockerfile

LICENSE

package.json

README.md

sql-boot-postgres-test.postm

config.xml

Dockerfile

```
39    <bean id="yamlProperties" class="org.springframework.beans.factory.config.  
40        <property name="resources" value="classpath:**/postgres/application-  
41    </bean>  
42    <context:property-placeholder properties-ref="yamlProperties"/>  
43  
44    <bean name="dev" id="dataSource" parent="templateCP" primary="true">  
45        <property name="driverClassName" value="org.postgresql.Driver" />  
46        <property name="url" value="${db_dev_url}" />  
47        <property name="username" value="${db_dev_user}" />  
48        <property name="password" value="${db_dev_password}" />  
49    </bean>  
50  
51    <import resource="table.xml" />  
52  
53
```

Project

postgres

.git

conf

application-postgres.yml

config.xml

table.xml

.travis.yml

docker-compose.yml

Dockerfile

LICENSE

package.json

README.md

sql-boot-postgres-test.postm

config.xml

```
35    <bean name="table,tbl,t" parent="dbResourceType">
36        <constructor-arg name="aliases" value="table,tbl,t" />
37        <constructor-arg name="child">
38            <list>
39                <value>#{column}</value>
40                <value>#{index}</value>-->
41                <value>#{pk}</value>
42                <value>#{fk}</value>
43                <value>#{trigger}</value>
44                <value>#{table_grant}</value>
45                <value>#{table_comment}</value>
46            </list>
47        </constructor-arg>
48        <constructor-arg name="readers">
49            <bean parent="sqlResourceReader">
50                <constructor-arg name="actionGenerator">
51                    <bean parent="plainTextTemplateGenerator" c:baseText="#{tables}">
52                </constructor-arg>
```

table.xml



Dockerfile

«Reader»

```
45          <value>#{table_comment}</value>
46      </list>
47  </constructor-arg>
48  <constructor-arg name="readers">
49      <bean parent="sqlResourceReader">
50          <constructor-arg name="actionGenerator">
51              <bean parent="plainTextTemplateGenerator" c:baseText="${tables}" />
52          </constructor-arg>
53      </bean>
```

«Reader»

```
1  tables: |  
2      select t.table_schema      as "schema"  
3            , t.table_name       as "table"  
4            , t.table_schema    as "@table_schema"  
5            , t.table_name     as "@table_name"  
6            , t.table_catalog  as "@table_catalog"  
7            , t.table_type     as "@table_type"  
8            , t.self_referencing_column_name as "@self_referencing_column_name"  
9            , t.reference_generation   as "@reference_generation"  
10           , t.user_defined_type_catalog as "@user_defined_type_catalog"  
11           , t.user_defined_type_schema  as "@user_defined_type_schema"    roles}" />  
12           , t.user_defined_type_name   as "@user_defined_type_name"  
13           , t.is_insertable_into    as "@is_insertable_into"  
14           , t.is_typed             as "@is_typed"  
15   from information_schema.tables t  
16 where t.table_type='BASE TABLE'  
17     and lower(t.table_schema) like '$schema'  
18     and lower(t.table_name) like '$table'
```

«Reader»

- SqlReader
- JdbcReader
- FileSystemReader
- RestReader
- GitReader
- HgReader
- SvnReader
- etc

«Generator»

```
21  table.sql.create: |
22    create table "$schema"."$table.name" (<%srv.get('column').each %>
23      !{it.name().padRight(20)}!{it.headers().data_type.toLowerCase()}!{it.headers().len
24    );
25
26  table.sql.drop: |
27    drop table $schema.$table.name;
```

«Generator»

- ❑ PlainTextGenerator
- ❑ FileBaseGenerator
- ❑ SqlGenerator

«Aggregator»

- TextAggregator
- JsonAggregator
- XmlAggregator
- HtmlAggregator
- ZipAggregator

Project	application-post... 60 "listen": "test", 61 "script": { 62 "type": "text/javascript", 63 "exec": [64 "tests[\"Status code is 200\"] = responseCode.code === 200;", 65 "", 66 "tests[\"Content-Type is application/json\"] = ", 67 "responseHeaders[\"Content-Type\"].has(\"application/json\");" 68] 69 } 70 } 71], 72 "request": { 73 "url": { 74 "raw": "http://localhost:8080/api/pk/bookings?type=json", 75 "protocol": "http", 76 "auth": {}, 77 "headers": {} 78 } 79 } 80 } 81 } 82 } 83 } 84 } 85 } 86 } 87 } 88 } 89 } 90 } 91 } 92 } 93 } 94 } 95 } 96 } 97 } 98 } 99 } 100 } 101 } 102 } 103 } 104 } 105 } 106 } 107 } 108 } 109 } 110 } 111 } 112 } 113 } 114 } 115 } 116 } 117 } 118 } 119 } 120 } 121 } 122 } 123 } 124 } 125 } 126 } 127 } 128 } 129 } 130 } 131 } 132 } 133 } 134 } 135 } 136 } 137 } 138 } 139 } 140 } 141 } 142 } 143 } 144 } 145 } 146 } 147 } 148 } 149 } 150 } 151 } 152 } 153 } 154 } 155 } 156 } 157 } 158 } 159 } 160 } 161 } 162 } 163 } 164 } 165 } 166 } 167 } 168 } 169 } 170 } 171 } 172 } 173 } 174 } 175 } 176 } 177 } 178 } 179 } 180 } 181 } 182 } 183 } 184 } 185 } 186 } 187 } 188 } 189 } 190 } 191 } 192 } 193 } 194 } 195 } 196 } 197 } 198 } 199 } 200 } 201 } 202 } 203 } 204 } 205 } 206 } 207 } 208 } 209 } 210 } 211 } 212 } 213 } 214 } 215 } 216 } 217 } 218 } 219 } 220 } 221 } 222 } 223 } 224 } 225 } 226 } 227 } 228 } 229 } 230 } 231 } 232 } 233 } 234 } 235 } 236 } 237 } 238 } 239 } 240 } 241 } 242 } 243 } 244 } 245 } 246 } 247 } 248 } 249 } 250 } 251 } 252 } 253 } 254 } 255 } 256 } 257 } 258 } 259 } 260 } 261 } 262 } 263 } 264 } 265 } 266 } 267 } 268 } 269 } 270 } 271 } 272 } 273 } 274 } 275 } 276 } 277 } 278 } 279 } 280 } 281 } 282 } 283 } 284 } 285 } 286 } 287 } 288 } 289 } 290 } 291 } 292 } 293 } 294 } 295 } 296 } 297 } 298 } 299 } 300 } 301 } 302 } 303 } 304 } 305 } 306 } 307 } 308 } 309 } 310 } 311 } 312 } 313 } 314 } 315 } 316 } 317 } 318 } 319 } 320 } 321 } 322 } 323 } 324 } 325 } 326 } 327 } 328 } 329 } 330 } 331 } 332 } 333 } 334 } 335 } 336 } 337 } 338 } 339 } 340 } 341 } 342 } 343 } 344 } 345 } 346 } 347 } 348 } 349 } 350 } 351 } 352 } 353 } 354 } 355 } 356 } 357 } 358 } 359 } 360 } 361 } 362 } 363 } 364 } 365 } 366 } 367 } 368 } 369 } 370 } 371 } 372 } 373 } 374 } 375 } 376 } 377 } 378 } 379 } 380 } 381 } 382 } 383 } 384 } 385 } 386 } 387 } 388 } 389 } 390 } 391 } 392 } 393 } 394 } 395 } 396 } 397 } 398 } 399 } 400 } 401 } 402 } 403 } 404 } 405 } 406 } 407 } 408 } 409 } 410 } 411 } 412 } 413 } 414 } 415 } 416 } 417 } 418 } 419 } 420 } 421 } 422 } 423 } 424 } 425 } 426 } 427 } 428 } 429 } 430 } 431 } 432 } 433 } 434 } 435 } 436 } 437 } 438 } 439 } 440 } 441 } 442 } 443 } 444 } 445 } 446 } 447 } 448 } 449 } 450 } 451 } 452 } 453 } 454 } 455 } 456 } 457 } 458 } 459 } 460 } 461 } 462 } 463 } 464 } 465 } 466 } 467 } 468 } 469 } 470 } 471 } 472 } 473 } 474 } 475 } 476 } 477 } 478 } 479 } 480 } 481 } 482 } 483 } 484 } 485 } 486 } 487 } 488 } 489 } 490 } 491 } 492 } 493 } 494 } 495 } 496 } 497 } 498 } 499 } 500 } 501 } 502 } 503 } 504 } 505 } 506 } 507 } 508 } 509 } 510 } 511 } 512 } 513 } 514 } 515 } 516 } 517 } 518 } 519 } 520 } 521 } 522 } 523 } 524 } 525 } 526 } 527 } 528 } 529 } 530 } 531 } 532 } 533 } 534 } 535 } 536 } 537 } 538 } 539 } 540 } 541 } 542 } 543 } 544 } 545 } 546 } 547 } 548 } 549 } 550 } 551 } 552 } 553 } 554 } 555 } 556 } 557 } 558 } 559 } 560 } 561 } 562 } 563 } 564 } 565 } 566 } 567 } 568 } 569 } 570 } 571 } 572 } 573 } 574 } 575 } 576 } 577 } 578 } 579 } 580 } 581 } 582 } 583 } 584 } 585 } 586 } 587 } 588 } 589 } 590 } 591 } 592 } 593 } 594 } 595 } 596 } 597 } 598 } 599 } 600 } 601 } 602 } 603 } 604 } 605 } 606 } 607 } 608 } 609 } 610 } 611 } 612 } 613 } 614 } 615 } 616 } 617 } 618 } 619 } 620 } 621 } 622 } 623 } 624 } 625 } 626 } 627 } 628 } 629 } 630 } 631 } 632 } 633 } 634 } 635 } 636 } 637 } 638 } 639 } 640 } 641 } 642 } 643 } 644 } 645 } 646 } 647 } 648 } 649 } 650 } 651 } 652 } 653 } 654 } 655 } 656 } 657 } 658 } 659 } 660 } 661 } 662 } 663 } 664 } 665 } 666 } 667 } 668 } 669 } 670 } 671 } 672 } 673 } 674 } 675 } 676 } 677 } 678 } 679 } 680 } 681 } 682 } 683 } 684 } 685 } 686 } 687 } 688 } 689 } 690 } 691 } 692 } 693 } 694 } 695 } 696 } 697 } 698 } 699 } 700 } 701 } 702 } 703 } 704 } 705 } 706 } 707 } 708 } 709 } 710 } 711 } 712 } 713 } 714 } 715 } 716 } 717 } 718 } 719 } 720 } 721 } 722 } 723 } 724 } 725 } 726 } 727 } 728 } 729 } 730 } 731 } 732 } 733 } 734 } 735 } 736 } 737 } 738 } 739 } 740 } 741 } 742 } 743 } 744 } 745 } 746 } 747 } 748 } 749 } 750 } 751 } 752 } 753 } 754 } 755 } 756 } 757 } 758 } 759 } 760 } 761 } 762 } 763 } 764 } 765 } 766 } 767 } 768 } 769 } 770 } 771 } 772 } 773 } 774 } 775 } 776 } 777 } 778 } 779 } 780 } 781 } 782 } 783 } 784 } 785 } 786 } 787 } 788 } 789 } 790 } 791 } 792 } 793 } 794 } 795 } 796 } 797 } 798 } 799 } 800 } 801 } 802 } 803 } 804 } 805 } 806 } 807 } 808 } 809 } 810 } 811 } 812 } 813 } 814 } 815 } 816 } 817 } 818 } 819 } 820 } 821 } 822 } 823 } 824 } 825 } 826 } 827 } 828 } 829 } 830 } 831 } 832 } 833 } 834 } 835 } 836 } 837 } 838 } 839 } 840 } 841 } 842 } 843 } 844 } 845 } 846 } 847 } 848 } 849 } 850 } 851 } 852 } 853 } 854 } 855 } 856 } 857 } 858 } 859 } 860 } 861 } 862 } 863 } 864 } 865 } 866 } 867 } 868 } 869 } 870 } 871 } 872 } 873 } 874 } 875 } 876 } 877 } 878 } 879 } 880 } 881 } 882 } 883 } 884 } 885 } 886 } 887 } 888 } 889 } 890 } 891 } 892 } 893 } 894 } 895 } 896 } 897 } 898 } 899 } 900 } 901 } 902 } 903 } 904 } 905 } 906 } 907 } 908 } 909 } 910 } 911 } 912 } 913 } 914 } 915 } 916 } 917 } 918 } 919 } 920 } 921 } 922 } 923 } 924 } 925 } 926 } 927 } 928 } 929 } 930 } 931 } 932 } 933 } 934 } 935 } 936 } 937 } 938 } 939 } 940 } 941 } 942 } 943 } 944 } 945 } 946 } 947 } 948 } 949 } 950 } 951 } 952 } 953 } 954 } 955 } 956 } 957 } 958 } 959 } 960 } 961 } 962 } 963 } 964 } 965 } 966 } 967 } 968 } 969 } 970 } 971 } 972 } 973 } 974 } 975 } 976 } 977 } 978 } 979 } 980 } 981 } 982 } 983 } 984 } 985 } 986 } 987 } 988 } 989 } 990 } 991 } 992 } 993 } 994 } 995 } 996 } 997 } 998 } 999 } 1000 } 1001 } 1002 } 1003 } 1004 } 1005 } 1006 } 1007 } 1008 } 1009 } 1010 } 1011 } 1012 } 1013 } 1014 } 1015 } 1016 } 1017 } 1018 } 1019 } 1020 } 1021 } 1022 } 1023 } 1024 } 1025 } 1026 } 1027 } 1028 } 1029 } 1030 } 1031 } 1032 } 1033 } 1034 } 1035 } 1036 } 1037 } 1038 } 1039 } 1040 } 1041 } 1042 } 1043 } 1044 } 1045 } 1046 } 1047 } 1048 } 1049 } 1050 } 1051 } 1052 } 1053 } 1054 } 1055 } 1056 } 1057 } 1058 } 1059 } 1060 } 1061 } 1062 } 1063 } 1064 } 1065 } 1066 } 1067 } 1068 } 1069 } 1070 } 1071 } 1072 } 1073 } 1074 } 1075 } 1076 } 1077 } 1078 } 1079 } 1080 } 1081 } 1082 } 1083 } 1084 } 1085 } 1086 } 1087 } 1088 } 1089 } 1090 } 1091 } 1092 } 1093 } 1094 } 1095 } 1096 } 1097 } 1098 } 1099 } 1100 } 1101 } 1102 } 1103 } 1104 } 1105 } 1106 } 1107 } 1108 } 1109 } 1110 } 1111 } 1112 } 1113 } 1114 } 1115 } 1116 } 1117 } 1118 } 1119 } 1120 } 1121 } 1122 } 1123 } 1124 } 1125 } 1126 } 1127 } 1128 } 1129 } 1130 } 1131 } 1132 } 1133 } 1134 } 1135 } 1136 } 1137 } 1138 } 1139 } 1140 } 1141 } 1142 } 1143 } 1144 } 1145 } 1146 } 1147 } 1148 } 1149 } 1150 } 1151 } 1152 } 1153 } 1154 } 1155 } 1156 } 1157 } 1158 } 1159 } 1160 } 1161 } 1162 } 1163 } 1164 } 1165 } 1166 } 1167 } 1168 } 1169 } 1170 } 1171 } 1172 } 1173 } 1174 } 1175 } 1176 } 1177 } 1178 } 1179 } 1180 } 1181 } 1182 } 1183 } 1184 } 1185 } 1186 } 1187 } 1188 } 1189 } 1190 } 1191 } 1192 } 1193 } 1194 } 1195 } 1196 } 1197 } 1198 } 1199 } 1200 } 1201 } 1202 } 1203 } 1204 } 1205 } 1206 } 1207 } 1208 } 1209 } 1210 } 1211 } 1212 } 1213 } 1214 } 1215 } 1216 } 1217 } 1218 } 1219 } 1220 } 1221 } 1222 } 1223 } 1224 } 1225 } 1226 } 1227 } 1228 } 1229 } 1230 } 1231 } 1232 } 1233 } 1234 } 1235 } 1236 } 1237 } 1238 } 1239 } 1240 } 1241 } 1242 } 1243 } 1244 } 1245 } 1246 } 1247 } 1248 } 1249 } 1250 } 1251 } 1252 } 1253 } 1254 } 1255 } 1256 } 1257 } 1258 } 1259 } 1260 } 1261 } 1262 } 1263 } 1264 } 1265 } 1266 } 1267 } 1268 } 1269 } 1270 } 1271 } 1272 } 1273 } 1274 } 1275 } 1276 } 1277 } 1278 } 1279 } 1280 } 1281 } 1282 } 1283 } 1284 } 1285 } 1286 } 1287 } 1288 } 1289 } 1290 } 1291 } 1292 } 1293 } 1294 } 1295 } 1296 } 1297 } 1298 } 1299 } 1300 } 1301 } 1302 } 1303 } 1304 } 1305 } 1306 } 1307 } 1308 } 1309 } 1310 } 1311 } 1312 } 1313 } 1314 } 1315 } 1316 } 1317 } 1318 } 1319 } 1320 } 1321 } 1322 } 1323 } 1324 } 1325 } 1326 } 1327 } 1328 } 1329 } 1330 } 1331 } 1332 } 1333 } 1334 } 1335 } 1336 } 1337 } 1338 } 1339 } 1340 } 1341 } 1342 } 1343 } 1344 } 1345 } 1346 } 1347 } 1348 } 1349 } 1350 } 1351 } 1352 } 1353 } 1354 } 1355 } 1356 } 1357 } 1358 } 1359 } 1360 } 1361 } 1362 } 1363 } 1364 } 1365 } 1366 } 1367 } 1368 } 1369 } 1370 } 1371 } 1372 } 1373 } 1374 } 1375 } 1376 } 1377 } 1378 } 1379 } 1380 } 1381 } 1382 } 1383 } 1384 } 1385 } 1386 } 1387 } 1388 } 1389 } 1390 } 1391 } 1392 } 1393 } 1394 } 1395 } 1396 } 1397 } 1398 } 1399 } 1400 } 1401 } 1402 } 1403 } 1404 } 1405 } 1406 } 1407 } 1408 } 1409 } 1410 } 1411 } 1412 } 1413 } 1414 } 1415 } 1416 } 1417 } 1418 } 1419 } 1420 } 1421 } 1422 } 1423 } 1424 } 1425 } 1426 } 1427 } 1428 } 1429 } 1430 } 1431 } 1432 } 1433 } 1434 } 1435 } 1436 } 1437 } 1438 } 1439 } 1440 } 1441 } 1442 } 1443 } 1444 } 1445 } 1446 } 1447 } 1448 } 1449 } 1450 } 1451 } 1452 } 1453 } 1454 } 1455 } 1456 } 1457 } 1458 } 1459 } 1460 } 1461 } 1462 } 1463 } 1464 } 1465 } 1466 } 1467 } 1468 } 1469 } 1470 } 1471 } 1472 } 1473 } 1474 } 1475 } 1476 } 1477 } 1478 } 1479 } 1480 } 1481 } 1482 } 1483 } 1484 } 1485 } 1486 } 1487 } 1488 } 1489 } 1490 } 1491 } 1492 } 1493 } 1494 } 1495 } 1496 } 1497 } 1498 } 1499 } 1500 } 1501 } 1502 } 1503 } 1504 } 1505 } 1506 } 1507 } 1508 } 1509 } 1510 } 1511 } 1512 } 1513 } 1514 } 1515 } 1516 } 1517 } 1518 } 1519 } 1520 } 1521 } 1522 } 1523 } 1524 } 1525 } 1526 } 1527 } 1528 } 1529 } 1530 } 1531 } 1532 } 1533 } 1534 } 1535 } 1536 } 1537 } 1538 } 1539 } 1540 } 1541 } 1542 } 1543 } 1544 } 1545 } 1546 } 1547 } 1548 } 1549 } 1550 } 1551 } 1552 } 1553 } 1554 } 1555 } 1556 } 1557 } 1558 } 1559 } 1560 } 1561 } 1562 } 1563 } 1564 } 1565 } 1566 } 1567 } 1568 } 1569 } 1570 } 1571 } 1572 } 1573 } 1574 } 1575 } 1576 } 1577 } 1578 } 1579 } 1580 } 1581 } 1582 } 1583 } 1584 } 1585 } 1586 } 1587 } 1588 } 1589 } 1590 } 1591 } 1592 } 1593 } 1594 } 1595 } 1596 } 1597 } 1598 } 1599 } 1600 } 1601 } 1602 } 1603 } 1604 } 1605 } 1606 } 1607 } 1608 } 1609 } 1610 } 1611 } 1612 } 1613 } 1614 } 1615 } 1616 } 1617 } 1618 } 1619 } 1620 } 1621 } 1622 } 1623 } 1624 } 1625 } 1626 } 1627 } 1628 } 1629 } 1630 } 1631 } 1632 } 1633 } 1634 } 1635 } 1636 } 1637 } 1638 } 1639 } 1640 } 1641 } 1642 } 1643 } 1644 } 1645 } 1646 } 1647 } 1648 } 1649 } 1650 } 1651 } 1652 } 1653 } 1654 } 1655 } 1656 } 1657 } 1658 } 1659 } 1660 } 1661 } 1662 } 1663 } 1664 } 1665 } 1666 } 1667 } 1668 } 1669 } 1670 } 1671 } 1672 } 1673 } 1674 } 1675 } 1676 } 1677 } 1678 } 1679 } 1680 } 1681 } 1682 } 1683 } 1684 } 1685 } 1686 } 1687 } 1688 } 1689 } 1690 } 1691 } 1692 } 1693 } 1694 } 1695 } 1696 } 1697
---------	--

Dockerfile

```
FROM mgramin/sql-boot
ADD conf conf/postgresql
RUN curl https://jdbc.postgresql.org/.../postgresql.jar
```

sql-boot-postgres-test.postman_collection.json update

README.md

sql-boot configuration for postgresql

build passing

```
691 sql-boot-postgres-test
692
693 □ index
694 ↴ get all indexes in json from schema
695     GET http://localhost:8080/api/index/bookings?type=json [200 OK, 1.83KB, 489ms]
696
697 □ pk
698 ↴ get all pk in json from schema
699     GET http://localhost:8080/api/pk/bookings?type=json [200 OK, 941B, 51ms]
700     ✓ Status code is 200
701     ✓ Content-Type is application/json
702
703 ↴ get all pk in xml from schema copy
704     GET http://localhost:8080/api/pk/bookings?type=xml [200 OK, 5.66KB, 266ms]
705     ✓ Status code is 200
706     ✓ Content-Type is application/json
```

```
708 □ table
709 ↴ get all tables ddl from schema
710   GET http://localhost:8080/api/table/bookings [200 OK, 1.6KB, 458ms]
711     ✓ Status code is 200
712     ✓ Content-Type is application/json
713
714 ↴ get all db tables ddl
715   GET http://localhost:8080/api/table [200 OK, 18.52KB, 2.4s]
716
717 ↴ get one table ddl
718   GET http://localhost:8080/api/table/bookings.flights [200 OK, 796B, 73ms]
719
720 ↴ get one table columns ddl
721   GET http://localhost:8080/api/column/bookings.flights [200 OK, 607B, 144ms]
722
```

```
708 □ table
709 ↴ get all t
710   GET http://
711   ✓ Status
712   ✓ Counter
713
714 ↴ get all d
715   GET http://
716
717 ↴ get one t
718   GET http://
719
720 ↴ get one t
721   GET http://
722
```

		executed	failed
723	iterations	1	0
724	requests	7	0
725	test-scripts	3	0
726	prerequest-scripts	0	0
727	assertions	6	0
728	total run duration:	4.2s	
729	total data received:	28.67KB (approx)	
730	average response time:	556ms	
731			
732			
733			
734			
735			
736			
737			
738			
739			
740			
741			

458ms]

796B, 73ms]

, 607B, 144ms]

My Repositories +

✓ sql-boot/oracle # 24

⌚ Duration: 3 min 23 sec

⌚ Finished: about an hour ago

✓ sql-boot/cassandra # 31

⌚ Duration: 1 min 48 sec

⌚ Finished: about an hour ago

✓ sql-boot/postgres # 74

⌚ Duration: 2 min 16 sec

⌚ Finished: about an hour ago

✓ mgramin/sql-boot # 250

⌚ Duration: 2 min 46 sec

⌚ Finished: about 3 hours ago

Current Branch

✓ master

⌚ Commit

⌚ Branch

⌚ mg

This job is part of our blog article

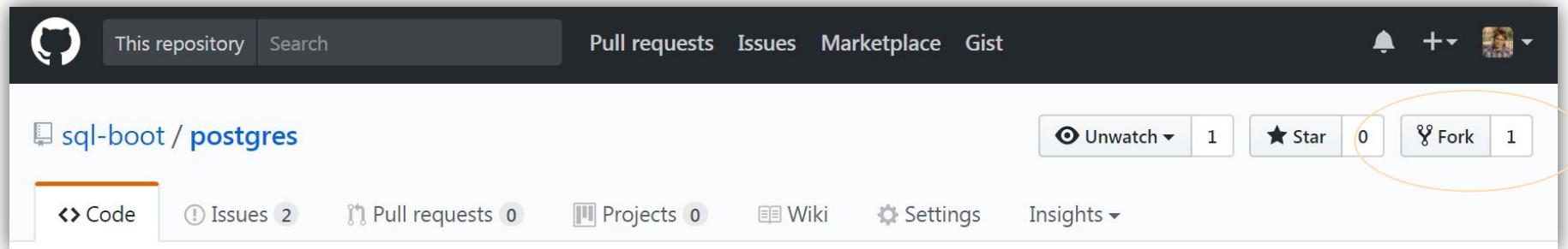
Jobs

▶ 1 Worker

▶ 6 Builds

474

Собственная конфигурация



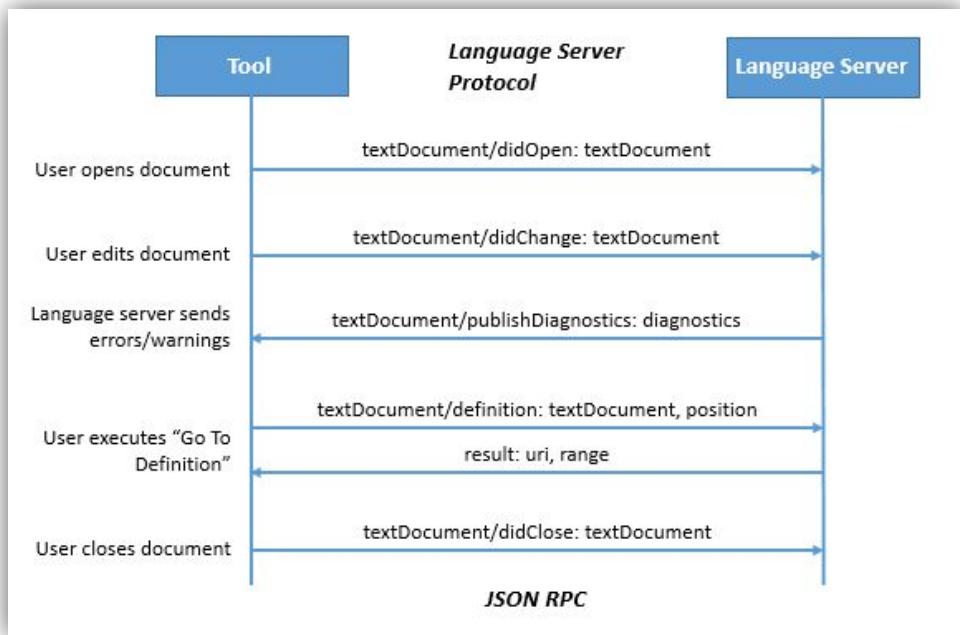
Forking sql-boot/postgres

It should only take a few seconds.

 Refresh



Перспективы



Language Server Protocol



Выводы

- Everything As a Code
- Transparency
- Open Source
- Community

Вопросы?



github.com/mgramin/sql-boot
mgramin@gmail.com