# Общее описание структуры таблиц PostgreSQL

PostgreSQL на платформе используется как дополнительная БД.

Данные разбиты по схемам. Схемы между собой независимы.

|  |  |
| --- | --- |
| Схема | Описание |
| boi | Предназначена для работы с инстанциями бизнес-объектов |
| calendar\_boi | Предназначена для работы с календарём |
| event\_trap | Тут записываются ловушки событий, например на ручное сохранение инстанции |
| person\_code | Тут работают с кодами пользователей |
| person\_upload | Тут загружаются пользователи |
| refresh\_bo | Здесь обновляются связи между БО |

## Схема boi:

|  |  |
| --- | --- |
| Таблица | Описание |
| [boi\_author](boi__boi_author.md) | Авторы инстанций бизнес-объектов |
| [boi\_authors](boi__boi_authors.md) | ??? |
| [boi\_ref](boi__boi_ref.md) | Сохраняет перекрёстные ссылки между бизнес-объектами |
| [department\_header](boi__department_header.md) | Связь департаментов с из руководителями |
| [node\_children](boi__node_children.md) | Список дочерних элементов каждой ноды |
| [node\_mirror](boi__node_mirror.md) | Зеркальное отображение структуры компании - в монго это PersonDto, DepartmentDto, CompanyDto |

## Схема calendar\_boi

|  |  |
| --- | --- |
| Таблица | Описание |
| [boi\_field\_change](calendar_boi__boi_field_change.md) | ???? |

## Схема event\_trap

|  |  |
| --- | --- |
| Таблица | Описание |
| [bo\_process\_schedule](event_trap__bo_process_schedule.md) | Запуск процессов по расписанию |
| [boi\_save](event_trap__boi_save.md) | Запуск процессов при сохранении инстанции через форму вручную |
| [delay\_timer](event_trap__delay_timer.md) | Запуск процессов по таймеру |
| [trap\_timer](event_trap__trap_timer.md) | Запуск процессов когда срабатывает ловушка на параллелограмм |

## Схема person\_code

|  |  |
| --- | --- |
| Таблица | Описание |
| [person\_entered\_code](person_code__person_entered_code.md) | Здесь храниться информация о количестве неверно введённых кодов подтверждения. Таблица нужна для блокировки пользователя, если он превысил некий лимит |

## Схема person\_upload

|  |  |
| --- | --- |
| Таблица | Описание |
| [person\_uploaded](person_upload__person_uploaded.md) | Здесь храниться информация о количестве порций файлов, которые были загружены в систему. Таблица нужна для временной блокировки возможности загрузки файлов для пользователя, если он превысил некий лимит |

## Схема refresh\_bo

|  |  |
| --- | --- |
| Таблица | Описание |
| [ref](refresh_bo__ref.md) | Здесь храниться информация о связях бизнес-объектов между собой. Таблица нужна для быстрого получения обратных связей |
| [ref\_removed](refresh_bo__ref_removed.md) | Здесь храниться информация о связях бизнес-объектов между собой. Таблица нужна для быстрого получения обратных связей |

# Описание таблицы boi.boi\_author

create table boi.boi\_author  
(  
 bo\_id varchar(20) not null,  
 bi\_id varchar(20) not null,  
 author\_id varchar(20),  
 actual boolean default true not null,  
 primary key (bo\_id, bi\_id)  
);  
  
comment on table boi\_author is 'Авторы инстансов бизнес-объектов';  
  
comment on column boi\_author.bo\_id is 'Идентификатор шаблона бизнес-объекта';  
  
comment on column boi\_author.bi\_id is 'Идентификатор инстанции бизнес-объекта';  
  
comment on column boi\_author.author\_id is 'Идентификатор автора инстанции';  
  
comment on column boi\_author.actual is 'Признак актуальности инстанции бизнес-объекта';

# Описание таблицы boi.boi\_authors

create table boi\_authors  
(  
 bo\_id varchar(20) not null,  
 author\_id varchar(20) not null,  
 primary key (bo\_id, author\_id)  
);

# Описание таблицы boi.boi\_\_boi\_ref

create table boi\_ref  
(  
 company\_id varchar(20) not null,  
 from\_instance\_id varchar(20) not null,  
 from\_bo\_id varchar(20) not null,  
 from\_field\_id varchar(20) not null,  
 target\_instance\_id varchar(20) not null,  
 target\_bo\_id varchar(20) not null,  
 from\_removed boolean default false not null,  
 target\_removed boolean default false not null,  
 primary key (from\_instance\_id, from\_field\_id, target\_instance\_id)  
);  
  
comment on table boi\_ref is 'Сохраняет перекрёстные ссылки между бизнес-объектами';  
  
comment on column boi\_ref.company\_id is 'Идентификатор компании';  
  
comment on column boi\_ref.from\_instance\_id is 'Идентификатор исходной инстанции';  
  
comment on column boi\_ref.from\_bo\_id is 'Идентификатор бизнес-объекта исходной инстанции';  
  
comment on column boi\_ref.from\_field\_id is 'Идентификатор поля исходной инстанции';  
  
comment on column boi\_ref.target\_instance\_id is 'Идентификатор инстанции назначения';  
  
comment on column boi\_ref.target\_bo\_id is 'Идентификатор бизнес-объекта инстанции назначения';  
  
comment on column boi\_ref.from\_removed is 'Признак удалённости исходной инстанции';  
  
comment on column boi\_ref.target\_removed is 'Признак удалённости инстанции назначения';

# Описание таблицы boi.department\_header

create table department\_header  
(  
 company\_id varchar(20) not null,  
 department\_id varchar(20) not null,  
 header\_person\_id varchar(20),  
 primary key (company\_id, department\_id)  
);  
  
comment on table department\_header is 'Связь департаментов с из руководителями';  
  
comment on column department\_header.company\_id is 'Идентификатор компании';  
  
comment on column department\_header.department\_id is 'Идентификатор департамента';  
  
comment on column department\_header.header\_person\_id is 'Идентификатор руководителя департамента';

# Описание таблицы boi.node\_children

create table node\_children  
(  
 company\_id varchar(20) not null,  
 node\_type varchar(15) not null,  
 node\_id varchar(20) not null,  
 child\_type varchar(15) not null,  
 child\_id varchar(20) not null,  
 primary key (company\_id, node\_id, child\_id)  
);  
  
comment on table node\_children is 'Список дочерних элементов каждой ноды';  
  
comment on column node\_children.company\_id is 'Идентификатор компании';  
  
comment on column node\_children.node\_type is 'Тип орг единицы';  
  
comment on column node\_children.node\_id is 'Идентификатор орг единицы';  
  
comment on column node\_children.child\_type is 'Тип дочерней орг единицы';  
  
comment on column node\_children.child\_id is 'Идентификатор дочерней орг единицы';

# Описание таблицы boi.node\_mirror

create table node\_mirror  
(  
 company\_id varchar(20) not null,  
 node\_type varchar(15) not null,  
 node\_id varchar(30) not null,  
 parent\_department\_id varchar(20),  
 primary key (company\_id, node\_id)  
);  
  
comment on table node\_mirror is 'Зеркальное отображение структуры компании - в монго это PersonDto, DepartmentDto, CompanyDto';  
  
comment on column node\_mirror.company\_id is 'Идентификатор компании';  
  
comment on column node\_mirror.node\_type is 'Тип орг единицы - OrgNodeType';  
  
comment on column node\_mirror.node\_id is 'Идентификатор орг единицы';  
  
comment on column node\_mirror.parent\_department\_id is 'Идентификатор родительского департамента';

# Описание таблицы calendar\_boi.boi\_field\_change

create table boi\_field\_change  
(  
 bo\_id varchar(32) not null,  
 boi\_id varchar(32) not null,  
 date\_field\_id varchar(32) not null,  
 status\_field\_id varchar(32) not null,  
 date\_field\_value varchar(200),  
 status\_field\_value varchar(200),  
 nom bigserial,  
 op\_id varchar(32),  
 op\_timestamp timestamp,  
 primary key (bo\_id, boi\_id, date\_field\_id, status\_field\_id)  
);

# Описание таблицы event\_trap.bo\_process\_schedule

create table bo\_process\_schedule  
(  
 id bigserial primary key,  
 bo\_process\_id varchar(20) not null,  
 run\_way\_id varchar(20) not null,  
 date\_time\_of\_event timestamp not null,  
 boi\_id varchar(20),  
 count\_of\_repeat numeric default 0 not null,  
 count\_of\_repeat\_on\_error numeric default 0 not null,  
 count\_of\_repeat\_on\_error\_increment\_at timestamp,  
 error\_log text,  
 done smallint default 0 not null,  
 actual smallint default 0 not null  
);

# Описание таблицы event\_trap.boi\_save

create table boi\_save  
(  
 id bigserial primary key,  
 done boolean default false not null,  
 bo\_id text not null,  
 boi\_id text not null,  
 bo\_process\_step\_id text not null,  
 created\_at timestamp default clock\_timestamp() not null,  
 actual smallint default 0 not null,  
 error\_log text,  
 bo\_process\_id varchar(20)  
);

# Описание таблицы event\_trap.boi\_save

create table delay\_timer  
(  
 id bigserial primary key,  
 date\_of\_event timestamp not null,  
 done smallint default 0 not null,  
 bo\_process\_step\_id text not null,  
 count numeric default 0 not null,  
 count\_increment\_at timestamp,  
 error\_log text,  
 created\_at timestamp default clock\_timestamp() not null,  
 actual smallint default 0 not null,  
 bo\_process\_id varchar(20),  
 active boolean default false  
);

# Описание таблицы event\_trap.trap\_timer

create table trap\_timer  
(  
 id bigserial primary key,  
 boi\_save\_id bigint not null references boi\_save,  
 arrow\_id text not null,  
 date\_of\_event timestamp not null,  
 done smallint default 0 not null,  
 count numeric default 0 not null,  
 count\_increment\_at timestamp,  
 error\_log text,  
 created\_at timestamp default clock\_timestamp() not null,  
 actual smallint default 0 not null,  
 active boolean default false  
);

# Описание таблицы person\_code.person\_entered\_code

create table person\_entered\_code  
(  
 id bigserial  
 primary key,  
 person\_id varchar(128) not null,  
 wrong\_entered\_code\_count integer default 0,  
 code\_type varchar(128) not null,  
 last\_entered\_at timestamp default clock\_timestamp(),  
 constraint person\_entered\_code\_person\_id\_code\_type\_uniq  
 unique (person\_id, code\_type)  
);  
  
comment on table person\_entered\_code is 'Здесь храниться информация о количестве неверно введённых кодов подтверждения. Таблица нужна для блокировки пользователя, если он превысил некий лимит';  
  
comment on column person\_entered\_code.person\_id is 'ИД пользователя, который неверно вводит коды подтверждения';  
  
comment on column person\_entered\_code.wrong\_entered\_code\_count is 'Количество неверно введённых кодов подтверждения';  
  
comment on column person\_entered\_code.code\_type is 'Тип подтверждения, который проходит пользователь';  
  
comment on column person\_entered\_code.last\_entered\_at is 'Момент когда пользователь последний раз ввёл код';

# Описание таблицы person\_upload.person\_uploaded

create table person\_uploaded  
(  
 id bigserial primary key,  
 person\_id varchar(128) not null constraint person\_uploaded\_person\_id unique,  
 upload\_count integer default 0,  
 last\_count\_reset\_at timestamp default clock\_timestamp()  
);  
  
comment on table person\_uploaded is 'Здесь храниться информация о количестве порций файлов, которые были загружены в систему. Таблица нужна для временной блокировки возможности загрузки файлов для пользователя, если он превысил некий лимит';  
  
comment on column person\_uploaded.person\_id is 'ИД пользователя, который производит загрузку файлов';  
  
comment on column person\_uploaded.upload\_count is 'Количество загруженных файлов до сброса счётчика';  
  
comment on column person\_uploaded.last\_count\_reset\_at is 'Момент когда был сброшен счётчик количества загруженных файлов';

# Описание таблицы refresh\_bo.ref

create table ref  
(  
 company\_id varchar(20) not null,  
 from\_bo\_id varchar(20) not null,  
 from\_instance\_id varchar(20) not null,  
 from\_field\_id varchar(20) not null,  
 to\_bo\_id varchar(20) not null,  
 to\_instance\_id varchar(20) not null,  
 primary key (to\_instance\_id, from\_instance\_id, from\_field\_id)  
);  
  
comment on table ref is 'Здесь храниться информация о связях бизнес-объектов между собой. Таблица нужна для быстрого получения обратных связей';  
  
comment on column ref.company\_id is 'ИД компании, в которой находятся ссылающиеся инстанции бизнес-объектов';  
  
comment on column ref.from\_bo\_id is 'ИД бизнес-объекта, инстанция которого ссылается на другую инстанцию';  
  
comment on column ref.from\_instance\_id is 'ИД инстанции, которая ссылается на другую инстанцию';  
  
comment on column ref.from\_field\_id is 'ИД поля, по которому инстанция ссылается на другую инстанцию';  
  
comment on column ref.to\_bo\_id is 'ИД бизнес-объекта инстанции, на которую ссылается инстанция from\_instance\_id';  
  
comment on column ref.to\_instance\_id is 'ИД инстанции, на которую ссылается инстанция from\_instance\_id';

# Описание таблицы refresh\_bo.ref\_removed

create table ref\_removed  
(  
 company\_id varchar(20) not null,  
 from\_bo\_id varchar(20) not null,  
 from\_instance\_id varchar(20) not null,  
 from\_field\_id varchar(20) not null,  
 to\_bo\_id varchar(20) not null,  
 to\_instance\_id varchar(20) not null,  
 primary key (to\_instance\_id, from\_instance\_id, from\_field\_id)  
);  
  
comment on table ref\_removed is 'Здесь храниться информация о связях бизнес-объектов между собой. Таблица нужна для быстрого получения обратных связей';  
  
comment on column ref\_removed.company\_id is 'ИД компании, в которой находятся ссылающиеся инстанции бизнес-объектов';  
  
comment on column ref\_removed.from\_bo\_id is 'ИД бизнес-объекта, инстанция которого ссылается на другую инстанцию';  
  
comment on column ref\_removed.from\_instance\_id is 'ИД инстанции, которая ссылается на другую инстанцию';  
  
comment on column ref\_removed.from\_field\_id is 'ИД поля, по которому инстанция ссылается на другую инстанцию';  
  
comment on column ref\_removed.to\_bo\_id is 'ИД бизнес-объекта инстанции, на которую ссылается инстанция from\_instance\_id';  
  
comment on column ref\_removed.to\_instance\_id is 'ИД инстанции, на которую ссылается инстанция from\_instance\_id';