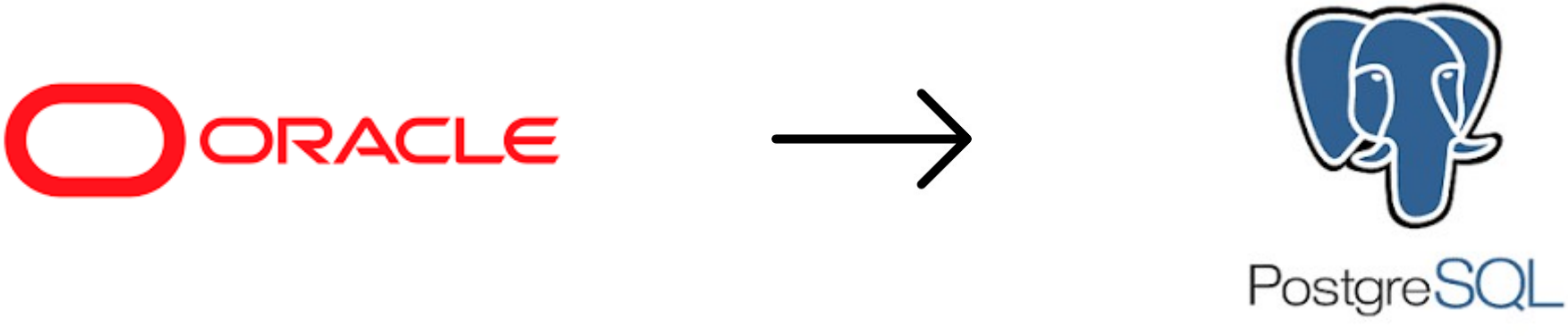
Организация непрерывной миграции и итоговой верификации базы данных для сложноорганизованной информационной системы и СУБД PostgreSQL



В.А. Роганов, М.А. Кривчиков

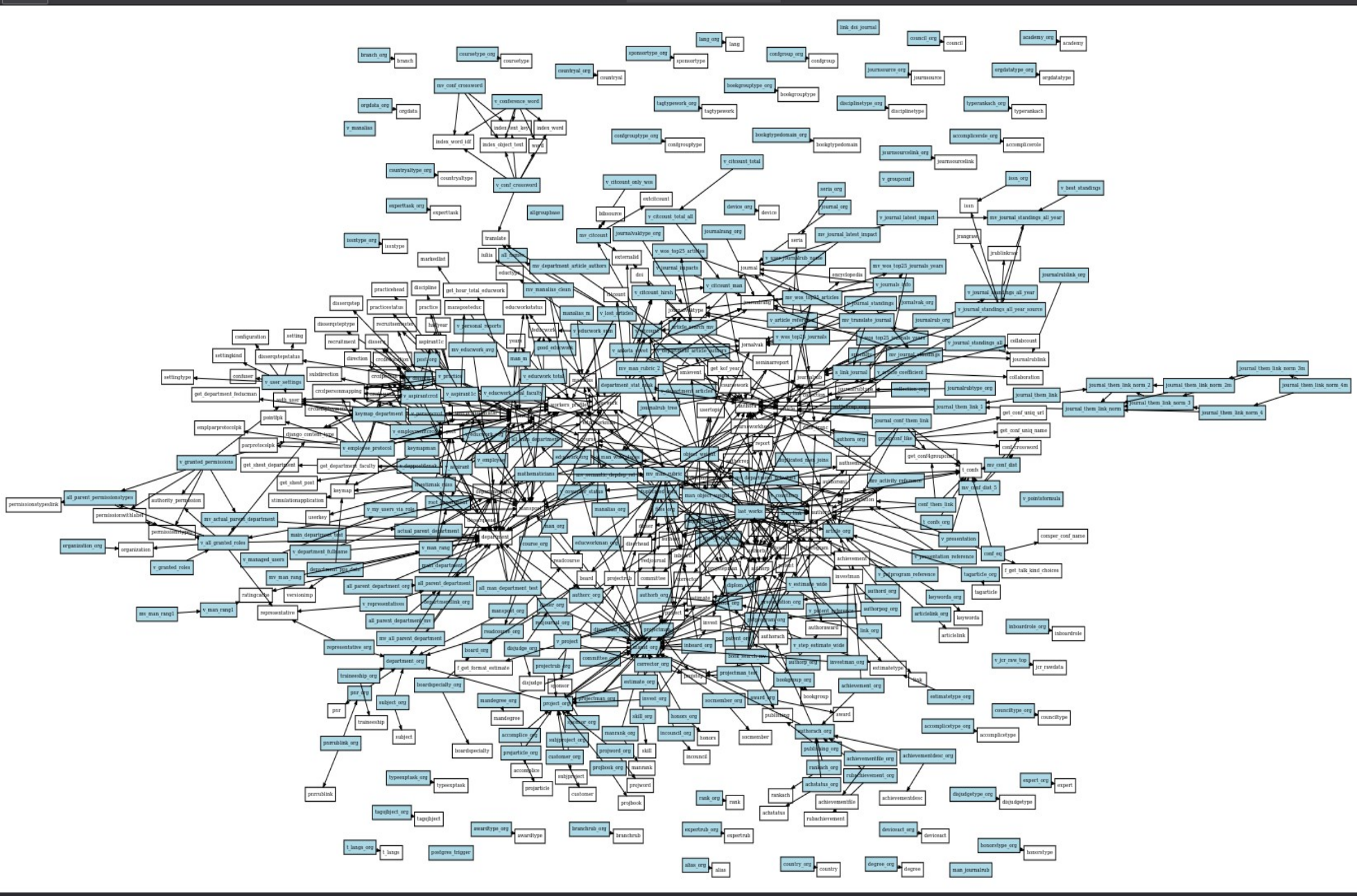
25 марта 2025 г., НИИ механики МГУ

Аннотация

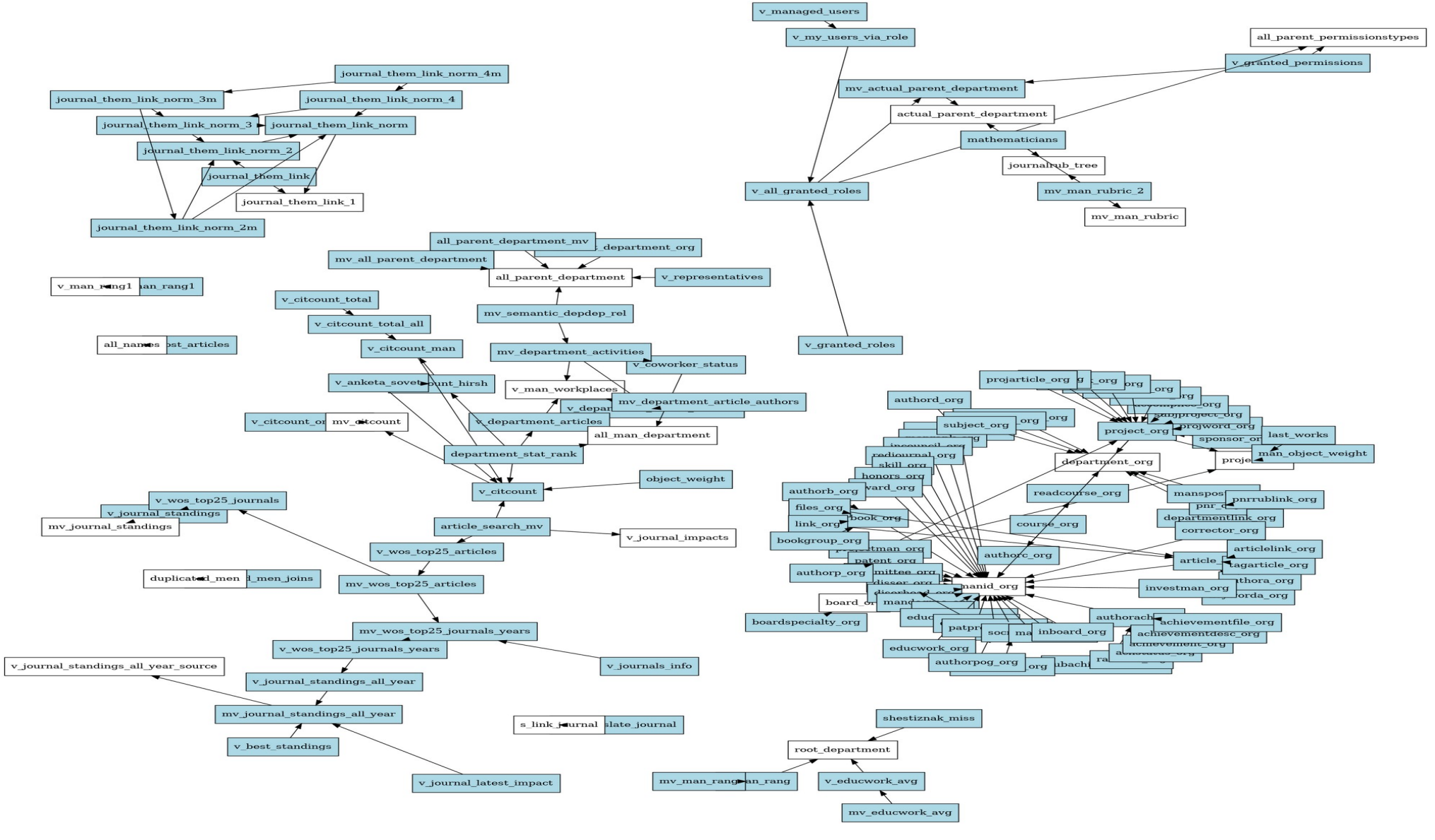
Миграция баз данных информационных систем с **богатой внутренней структурой** на новую СУБД является нетривиальной задачей, для решения которой не существует универсального механизма.

В докладе рассматривается набор адаптированных существующих и дополнительно разработанных средств, использованных в рамках миграции БД информационной наукометрической системы "ИСТИНА" с СУБД Oracle на СУБД PostgreSQL.

При разработанном подходе подготовка к переходу становится частью эволюции системы, что снижает риски и позволяет лучше адаптироваться к особенностям новой СУБД.


link_doi_journal
council_org
council
academy_org
academy
lang_org
lang
branch_org
branch
coursetype_org
coursetype
sponsortype_org
sponsortype
configroup_org
configroup
countrysl_org
countrysl
journsource_org
journsource
orgdatatype_org
orgdatatype
mv_conf_crossword
v_conference_word
bookgrouptype_org
bookgrouptype
orgdata_org
orgdata
tagtypework_org
tagtypework
disciplinetype_org
disciplinetype
typerankach_org
typerankach
v_manalias
index_text_key
index_word
index_word_idf
index_object_text
word
configgrouptype_org
configgrouptype
bookgtypedomain_org
bookgtypedomain
accomplicerole_org
accomplicerole
journsourcelink_org
journsourcelink
countryaltype_org
countryaltype
v_conf_crossword
v_citcount_total
v_groupconf
issn_org
v_best_standings
expertask_org
expertask
allgroupbase
v_citcount_only_wos
selfcitcount
issntype_org
issntype
v_citcount_total_all
device_org
device
serta_org
journal_org
issn
v_journal_latest_impact
mv_journal_standings_all_year
translate
mv_citcount
journalvaktype_org
journalrang_org
serta
mv_journal_latest_impact
irangrank
markedlist
v_wos_top25_articles
v_user_journalrub_count
jrubrlinkwos
mv_department_article_authors
externalid
doi
v_journal_impacts
journal
encyclopedia
mv_wos_top25_journals_years
journalrublink_org
practicehead
discipline
get_hour_total_educwork
mv_manalias_clean
v_citcount_man
v_journals_info
v_journal_standings_all_year
dissertstep
practicestatus
practice
manegeeduc
educworkstatus
v_citcount_hirsh
mv_wos_top25_articles
journal_standings
journalvak_org
v_journal_standings_all_year_source
dissertsteptype
recruitmentstatus
manalias_m
v_lost_articles
v_article_rating
mv_translate_journal
journalrub_org
configuration
setting
recruitment
dissert
aspirant1c
v_personal_reports
v_educwork_stat
v_wos_top25_journals
mv_wos_top25_journals_years
v_journal_standings_all
collaboration_org
settingkind
dissertstepstatus
direction
mv_educwork_org
good_educwork
man_m
v_journal_standings
journalrublink
settingtype
confuser
subdirection
v_user_settings
v_practice
v_educwork_total
mv_man_rubric_2
get_kol_years
seminarreport
v_link_journal
v_article_coefficient
collaboration
get_department_feducman
v_aspirant1cfaculty
v_aspirant1c
v_educwork_total_faculty
department_stat_data
v_department_articles
journalrubtype_org
collection_org
journal_them_link
journal_them_link_norm_2
journal_them_link_norm_3m
journal_them_link_norm_2m
journal_them_link_norm_4m
journal_them_link_norm
journal_them_link_norm_3
journal_them_link_norm_4
journal_them_link_1
get_conf_uniq_url
pointfpk
keymap_department
journalrub_tree
journal_conf_them_link
get_conf_uniq_name
emplparprotocolpk
v_employee_protocol
keymapman
authors_org
conf_crossword
parprotocolpk
v_granted_permissions
get_sheet_department
get_department_faculty
v_employee
mathematicians
mv_conf_dist
mv_activity_reference
get_sheet_post
keymap
mv_conf_dist_5
permissionstypeslink
all_parent_permissionstypes
authority_permission
stimulationsapplication
conf_them_link
v_pointsformula
permissionwithlabel
mv_actual_parent_department
userkey
v_my_users_via_role
t_confs_org
permissionstypes
main_department_full
actual_parent_department
v_presentation
comper_conf_name
organization_org
organization
v_all_granted_roles
v_department_fullname
v_man_rang
v_presentation_reference
conf_eq
v_granted_roles
v_managed_users
v_pp_program_reference
f_get_talk_kind_choices
mv_man_rang
all_parent_department_org
all_parent_department
all_man_department_test
v_estimate_wide
investman
taqarticle
rating_org
versionimp
v_representatives
keywords_org
mv_man_rang1
v_man_rang1
representative
authorsd_org
authorpog_org
articlelink_org
keywords
all_parent_department_mv
authorsearch
articlelink
inboundrole_org
mv_all_parent_department
v_project
link_org
inboundrole
representative_org
v_jcr_raw_top
jcr_rawdata
department_org
v_step_estimate_wide
traineeship_org
f_get_format_estimate
estimatetype
estimatetype_org
boardspecialty_org
achievement_org
per_org
subject_org
mandegree_org
publishing
counciltype_org
counciltype
per
traineeship
mandegree
accomplicetype_org
accomplicetype
perrublink_org
subject
publishing_org
achievementfile_org
achievementdesc_org
rubachievement_org
rankach_org
typesofptask_org
typesofptask
achstatus_org
expert_org
expert
perrublink
rank_org
rank
achievementdesc
disjudgetype_org
disjudgetype
taqobject_org
taqobject
achievementfile
rubachievement
achstatus
deviceact_org
deviceact
swardtype_org
swardtype
branchrub_org
branchrub
expertrub_org
expertrub
honorstype_org
honorstype
t_langs_org
t_langs
postgres_trigger
alias_org
alias
country_org
country
degree_org
degree
man_journalrub

Зависимости для представлений



Доклад содержит некоторые материалы ко второй нашей статье.

Первая статья по миграции:

Кривчиков М.А., Роганов В.А., Васенин В.А.
Опыт миграции информационно-аналитической системы больших наукометрических данных на систему управления PostgreSQL
Программная Инженерия: Том: 15 Номер: 2 Год издания: 2024.

План доклада

Часть 1 (интересное)

- Инструменты
- Возможности СУБД
- Производительность
- Безопасность

Часть 2 (процесс)

- Подготовка перехода
- Верификация
- Этапы перехода
- Сопровождение

Инструменты

1. oracle container (development server)
2. ora-imp (импорт входного дампа)
3. column_types (к моделям Django)
4. ora2pg (правки ошибок): конвертация
5. pg-load (наложение схемы на данные)
6. schemacrawler (интроспекция Oracle & PG)
7. migra (правки ошибок) + интроспекция PG
8. compactify (по теме есть готовый текст)
9. db_diff (сравнение/мониторинг содержимого)
10. pgdev (многоцелевой скрипт для запуска PG)

Возможности СУБД PostgreSQL

Значительное количество различных расширений для языка SQL.
Универсальные графические клиенты: PgAdmin (Astra), DBeaver.

Два вида дампа/схемы БД:

- с пустыми данными для таблиц;
- без данных для таблиц.

Первый используется в качестве шаблона для наложения на данные.

Можно бесконечно искать дополнительные модули и инструменты !

Для PostgreSQL мало «родных» сервисных инструментов; их производством занимается огромное количество пользователей и разработчиков приложений для этой СУБД.

Производительность

Проблемы с производительностью СУБД PostgreSQL — да, они есть. Но нередко PG очень быстрый.

Замечания о привязке к специфике СУБД — это неизбежно для достижения максимальной производительности.

Технология: динамическая генерация запросов в функциях.

Плагины и встроенные средства для профилирования.

Физическая репликация данных (RO-реплики).

Безопасность

В качестве операционной системы - **Astra Linux** (отечественная разработка на базе Debian 12).

Запросы от "левых" пользователях - на RO-репликах.

В перспективе: внедрение защищенной версии БД (между версиями БД - логическая репликация данных).

Подготовка перехода

Глобально: три стадии перехода:

1. Добиваемся, чтобы все старые тесты работали;
2. Проходим верификацию и новые тесты (уровня БД);
3. Делаем переход и осуществляем активную поддержку.

Полезное из "круглого стола по миграции" — нужно **тщательнее следить** за тем, чтобы все сущности из оригинальной БД достигли целевой БД.

Извлечение **типов столбцов из моделей Django** — из-за строгой типизации в случае СУБД PostgreSQL разборки с типами столбцов — **отдельная заметная** часть работ по миграции.

Компактификация БД (10% и 1% полного объема) — крайне полезная вещь.

Подготовка перехода (продолжение)

Итог подготовки:
получение шаблона **pg-tmpl.sql** (~ 85.000 строк кода).

Удаление лишнего: 1000+ => 600+ таблиц (!).

Итерации: перенос данных - проверка типов столбцов.

Замена "лишних" триггеров на последовательности.

Реестр управляющих структур:
Разложенные по каталогам управляющие сущности.

Верификация

Мы с нуля (другими инструментами) проверяем все:

- по составу;
- по функциональности.

По составу: удобно использовать инструмент **schemacrawler**.

Новые тесты по функциональности:

Тесты по корректному отображению страниц;
Тесты на CRUD-операции для моделей Django;
Тесты уровня БД (упорядоченные по зависимостям);
Очная ставка для тестов уровня БД (частично).

Верификация **данных** после их переноса.

Этапы перехода

Одна из возможных схем по реализации «плавного» перехода:

1. Вначале запускаем/синхронизируем RO-копию системы.
2. Далее — размораживаем копию и просим энтузиастов с ней поработать.
3. Замораживаем оригинал, переходим полностью на PostgreSQL-версию.

	Ora	PG	
1.	RW*	RO	(не требует остановки)
2.	RW*	RW	(не требует остановки)
3.	RO	RW*	(заморозка оригинала)

Сопровождение

В процессе развития системы будет меняться и схема БД.

При этом все изменения вносятся и обкатываются вначале на стенде (так называемая **тестовая БД**), а уже потом переносятся в реальную рабочую базу.

Для инкрементного переноса изменений может использоваться **Migra**.

Для перехода на новые версии PostgreSQL (и в прочих случаях глобальных изменений) может быть задействован инструмент **pg-load**.