

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Факультет программной инженерии и компьютерной техники

**КУРСОВАЯ РАБОТА
ПО ДИСЦИПЛИНЕ «ХРАНИЛИЩА И БАЗЫ ДАННЫХ»**

Студенты:

Белоголов М. А.

Быков К. А.

Группа: Р41142

Преподаватель: Королёва Юлия Александровна

Санкт-Петербург

2020

Задание 1

Создать 4 БД, нормализовать их, где это необходимо. Возможно внесение дополнительных полей.

Запуск из корня проекта:

```
/usr/bin/python3 -m lab1 create
```

Повторный запуск удвоит количество записей во всех БД. Количество генерируемых записей можно изменять в функциях `fill_<database>` в `generator/<database>.py`

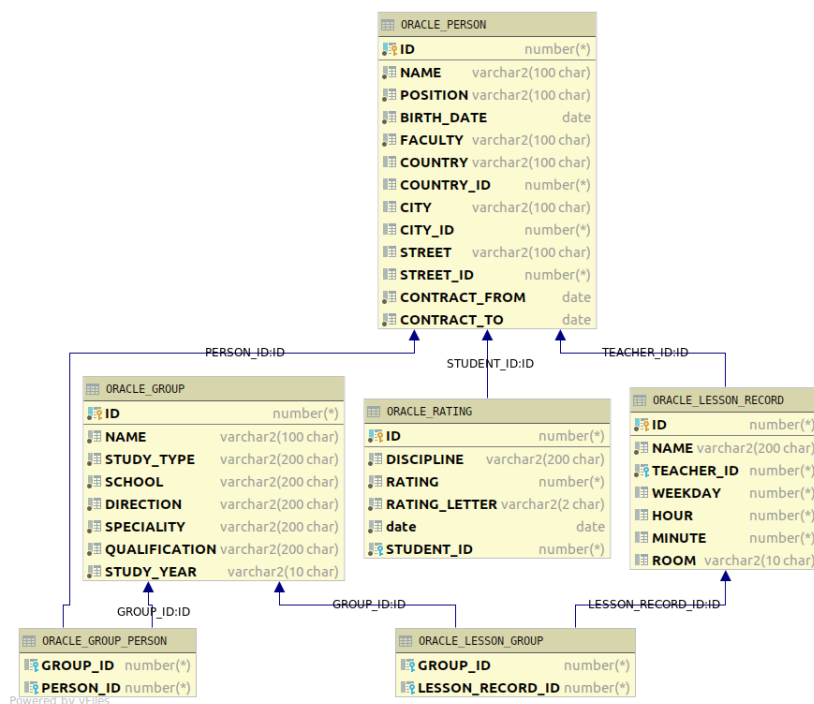


Figure 1. Oracle

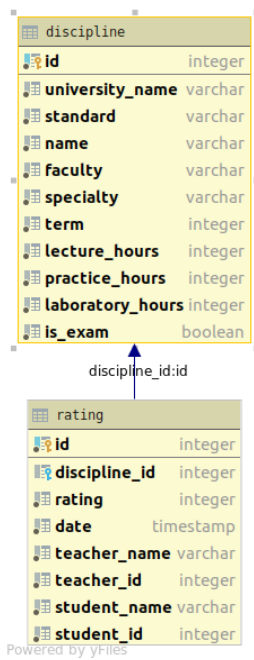


Figure 2. PostgreSQL

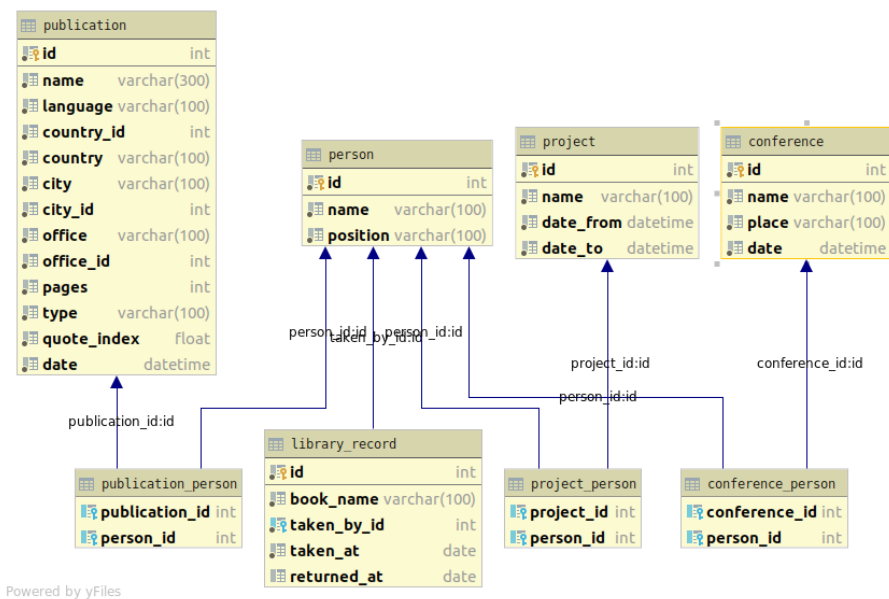


Figure 3. MySQL

Задание 2

Из четырёх созданных СУБД необходимо загрузить данные в одну итоговую схему БД (на Oracle).

Необходимо провести миграцию из четырёх БД в единую схему в Oracle DB. Объекты схемы можно найти в model/final.

Запуск:

`/usr/bin/python3 -m lab1 migrate`

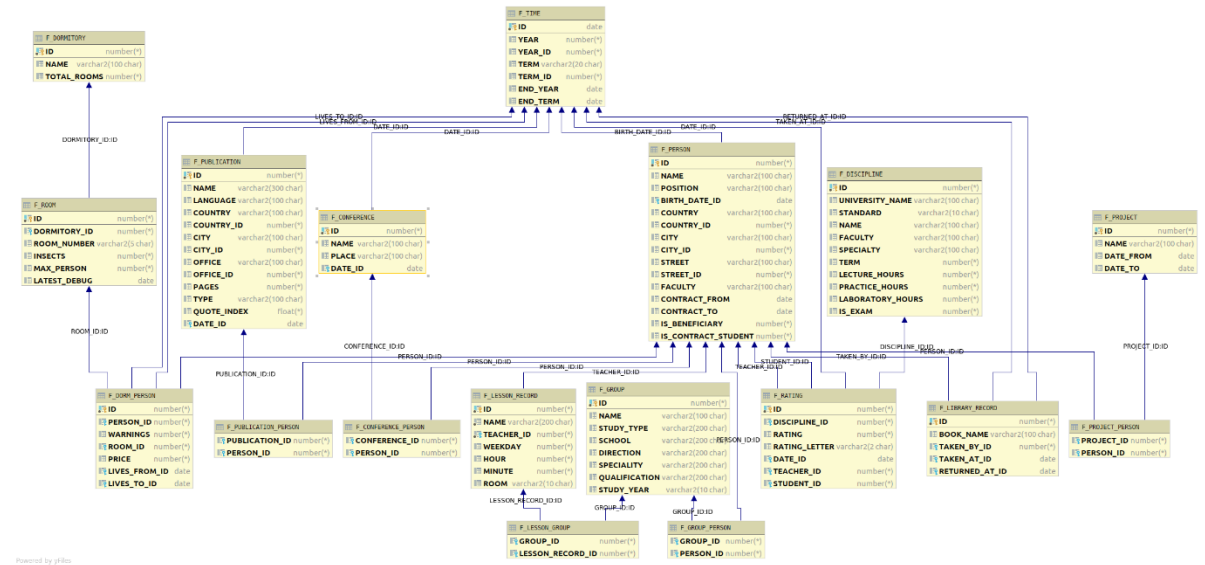


Figure 4. Схема

Задание 3

1. Ознакомиться с выложенной схемой
2. Заполнить схему данными из полученного во втором задании хранилища
3. На основе заполненной схемы создать рабочее место аналитика (AWM), куб, атрибуты и измерения для куба.
4. Получить несколько отчётов

Перед началом работы нужно создать таблицы фактов.

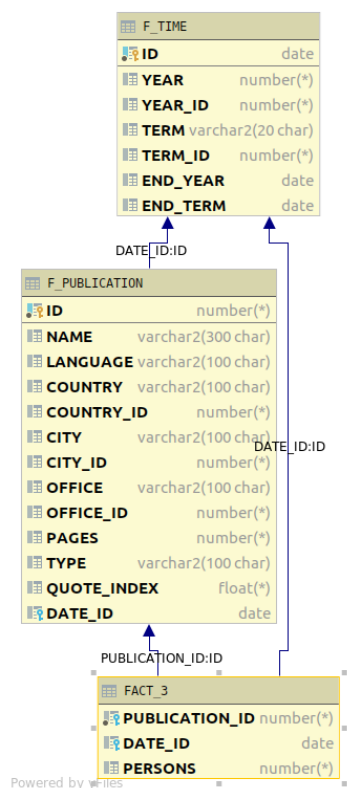


Figure 5. Таблица фактов

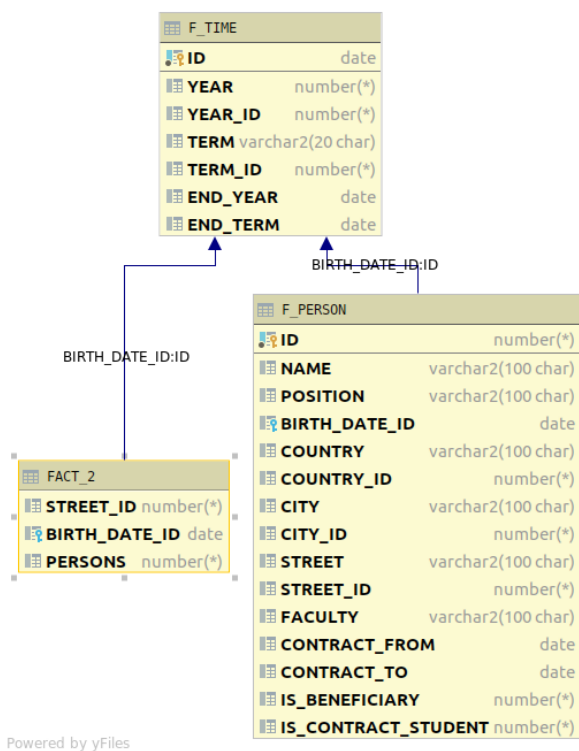


Figure 6. Таблица фактов

Необходимо получить из БД пару отчётов с помощью Oracle Analytic Workspace Manager. Подключаемся к нашей БД, создаём рабочее место аналитика.

Элементы или объекты в логической схеме. В данном задании они заданы преподавателем и обозначены в файле. В нашем случае их четыре:

- место рождения (birth_place)
- место публикации (publication_place)
- время (time)
- общежитие (dormitory)

Все данные можно группировать по различным критериям. Если грубо, то уровни и есть эти самые критерии. Проще всего это понять по времени. Данные со временем можно группировать по дням, неделям, месяцам, кварталам, года, etc. "День", "Неделя", "Месяц" -- это уровни Последовательность "День -> Неделя -> Месяц" -- это иерархия, где "день" -- это наиболее низкий уровень, а "месяц" - - наиболее высокий.

Уровни и их иерархии так же заданы в [файле](./res/er-rus for awm.pdf).

Уровни и иерархии:

- место рождения (birth_place)
 - все страны (all_countries)
 - страна (country)
 - город (city)
 - район (district)
- место публикации (publication_place)
 - все страны (all_countries)
 - страна (country)
 - город (city)
 - издание (office)
- время (time)
 - все года (all_years)
 - год (year)
 - семестр (term)

- общежитие (dormitory)
 - все общежития (all_dormitories)
 - общежитие (dormitory)

Заходим в маппинг величины и размечаем для каждого пункта свой столбец в таблице.

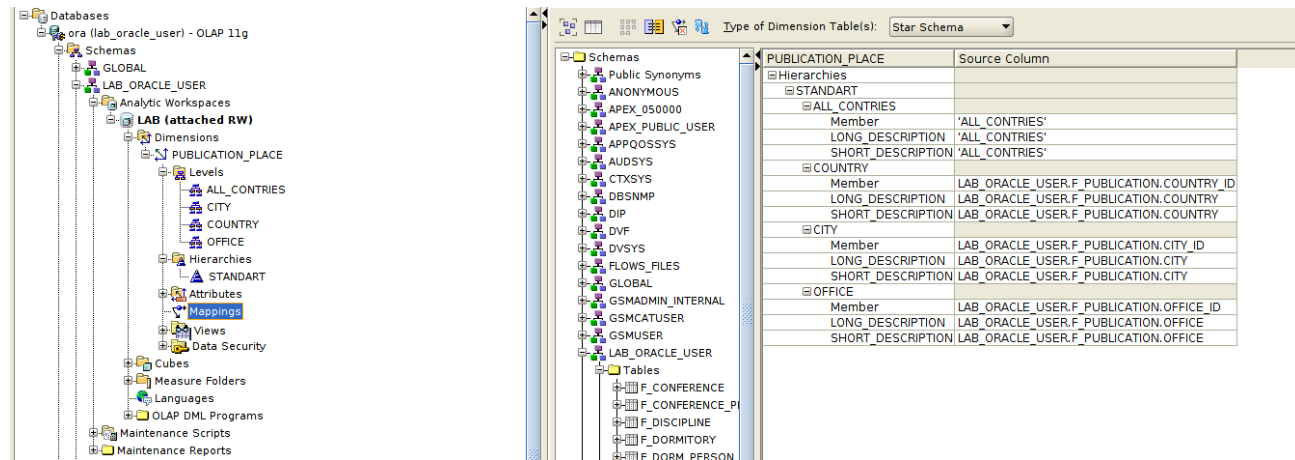
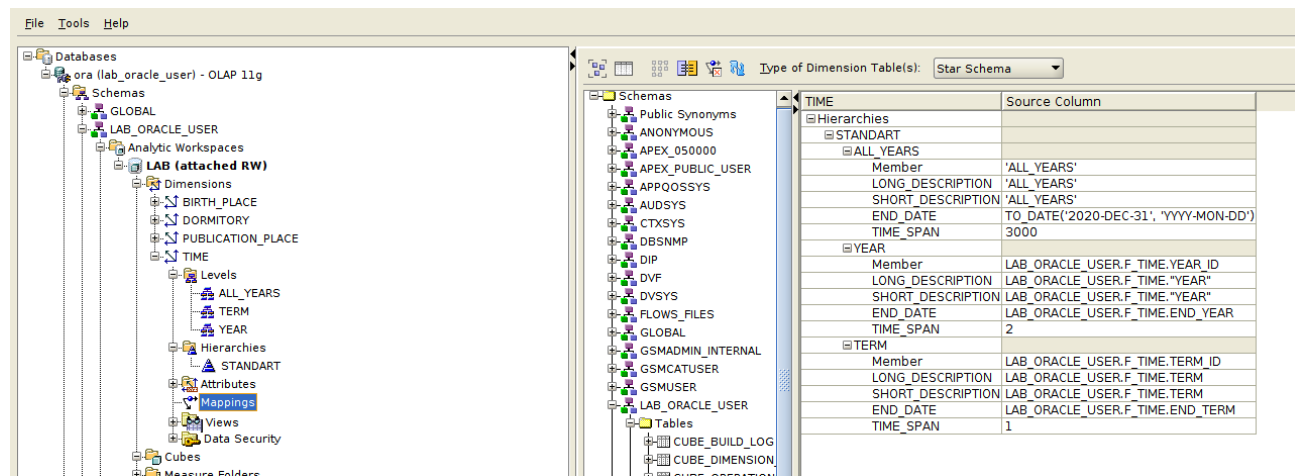


Figure 7. Mapping



Кубы - это представления данных на базе величин. Каждому кубу сопоставим свою таблицу фактов (по номеру).

LAB_CUBE_2

- Dimensions: time, birth_place
- Measures: TOTAL_PERSONS
- Calculated Measures: None

LAB_CUBE_3

- Dimensions: time, publication_place
- Measures: TOTAL_PERSONS
- Calculated Measures: None

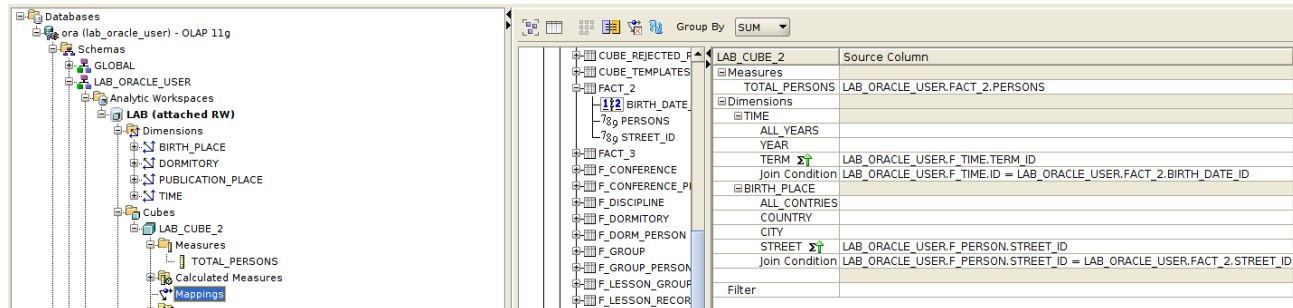
При создании куба во вкладке Partitioning **убираем** галку с пункта Partition cube.

После создания хотя бы одного измерения (Measures) можно размапить куб на данные. Если информация обо всех уровнях иерархии хранится в одной таблице, достаточно указать ссылку на нижний уровень.

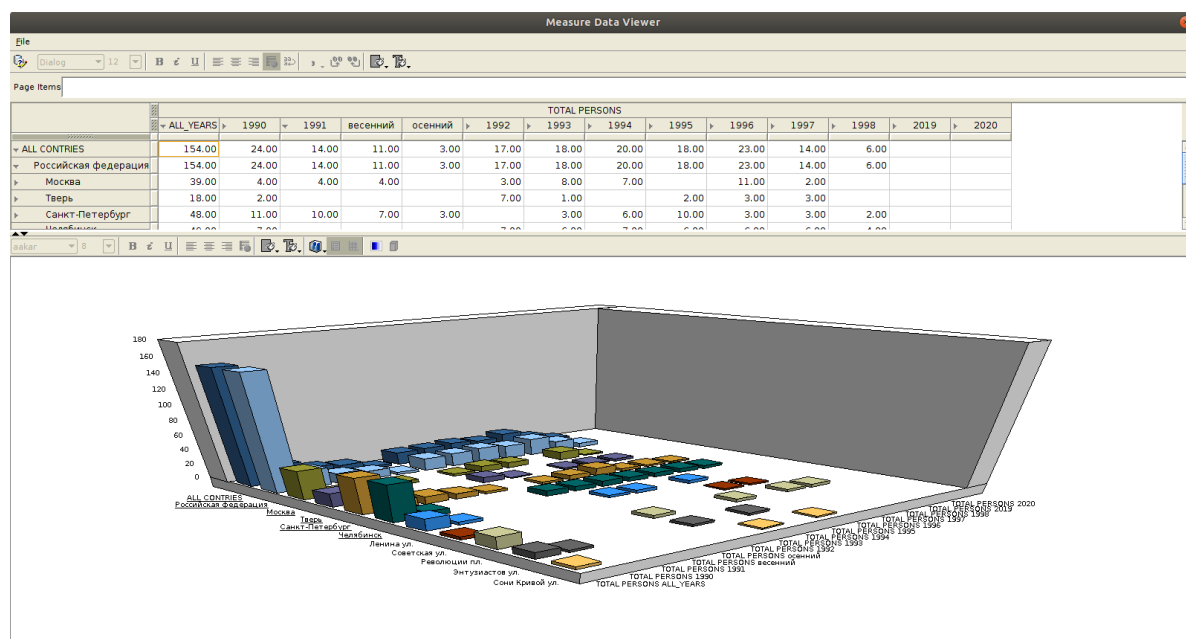
Так как мы считаем сумму людей, в верхней панели в селекторе *Group By* оставляем SUM.

В отличие от маппинга уровней иерархии, здесь нужно прописать Join Condition, который связывает таблицу фактов и таблицу с информацией. Предполагается, что таблица фактов - центр топологии "звезда" и Join можно прописать так:

<field1> = <field2>



Если всё сделано правильно, то нужно нажать правой кнопкой мыши по кубу -> *Maintain Cube*.



Вывод

В процессе данной лабораторной работы были подняты 4 базы данных, в них созданы схемы для хранения данных согласно заданию. Затем была создана схема, которая агрегирует данные из всех БД в одну и написано программное решение для автоматической миграции. На финальной схеме была проведена аналитика.