# **Проектирование Data Warehouse для системы образования (основа - университет)**

## 1. Бизнес-процесс

Для проектирования хранилища данных была выбрана область **академической успеваемости студентов**. В рамках университетской базы одним из ключевых процессов является выставление оценок студентам за посещённые занятия (лекции, практики, лабораторные работы).

Данный бизнесс-процесс выбран по следующим причинам:

1. Анализ оценок — это важный показатель качества обучения.
2. В базе уже есть все необходимые сущности: студенты, группы, занятия, преподаватели.
3. На этих данных легко строить разные отчёты (средние баллы, количество студентов, отличники).
4. Процесс универсален и подходит для всех университетов, факультетов и курсов.

## 2. Уровень детализации (Grain)

**Одна строка факта = одна оценка конкретного студента за конкретное занятие в определённое время.**

## 3. Таблицы измерений (dimension tables)

### 1. dim\_student — студенты

■ student\_sk: BIGINT, PK, NOT NULL

■ source\_student\_id: BIGINT, NOT NULL (ID из OLTP)

■ first\_name: VARCHAR(100), NOT NULL

■ last\_name: VARCHAR(100), NOT NULL

■ gender: CHAR(1), NOT NULL (M/F)

■ birth\_year: INT, NULL

■ current\_group\_name: VARCHAR(50), NULL

### 2. dim\_lesson — занятия

■ lesson\_sk: BIGINT, PK, NOT NULL

■ source\_lesson\_id: BIGINT, NOT NULL

■ subject\_name: VARCHAR(100), NOT NULL

■ classroom\_number: VARCHAR(20), NULL

■ building\_name: VARCHAR(100), NULL

■ faculty\_name: VARCHAR(100), NULL

■ university\_name: VARCHAR(150), NULL

### 3. dim\_teacher — преподаватели

■ teacher\_sk: BIGINT, PK, NOT NULL

■ source\_teacher\_id: BIGINT, NOT NULL

■ first\_name: VARCHAR(100), NOT NULL

■ last\_name: VARCHAR(100), NOT NULL

■ position: VARCHAR(50), NULL

■ department: VARCHAR(100), NULL

■ faculty\_name: VARCHAR(100), NULL

### 4. dim\_group — студенческие группы

■ group\_sk: BIGINT, PK, NOT NULL

■ source\_group\_id: BIGINT, NOT NULL

■ group\_name: VARCHAR(50), NOT NULL

■ curator\_teacher\_name: VARCHAR(150), NULL

■ start\_year: INT, NULL

## 4. Таблица фактов (fact table)

### fact\_student\_lesson

■ fact\_id: BIGINT, PK, NOT NULL

■ student\_sk: BIGINT, FK (= dim\_student(student\_sk), NOT NULL)

■ lesson\_sk: BIGINT, FK (= dim\_lesson(lesson\_sk), NOT NULL)

■ teacher\_sk: BIGINT, FK (= dim\_teacher(teacher\_sk), NOT NULL)

■ group\_sk: BIGINT, FK (= dim\_group(group\_sk), NOT NULL)

■ event\_time: TIMESTAMP, NOT NULL (дата и время занятия/оценки)

■ grade\_value: INT, NOT NULL

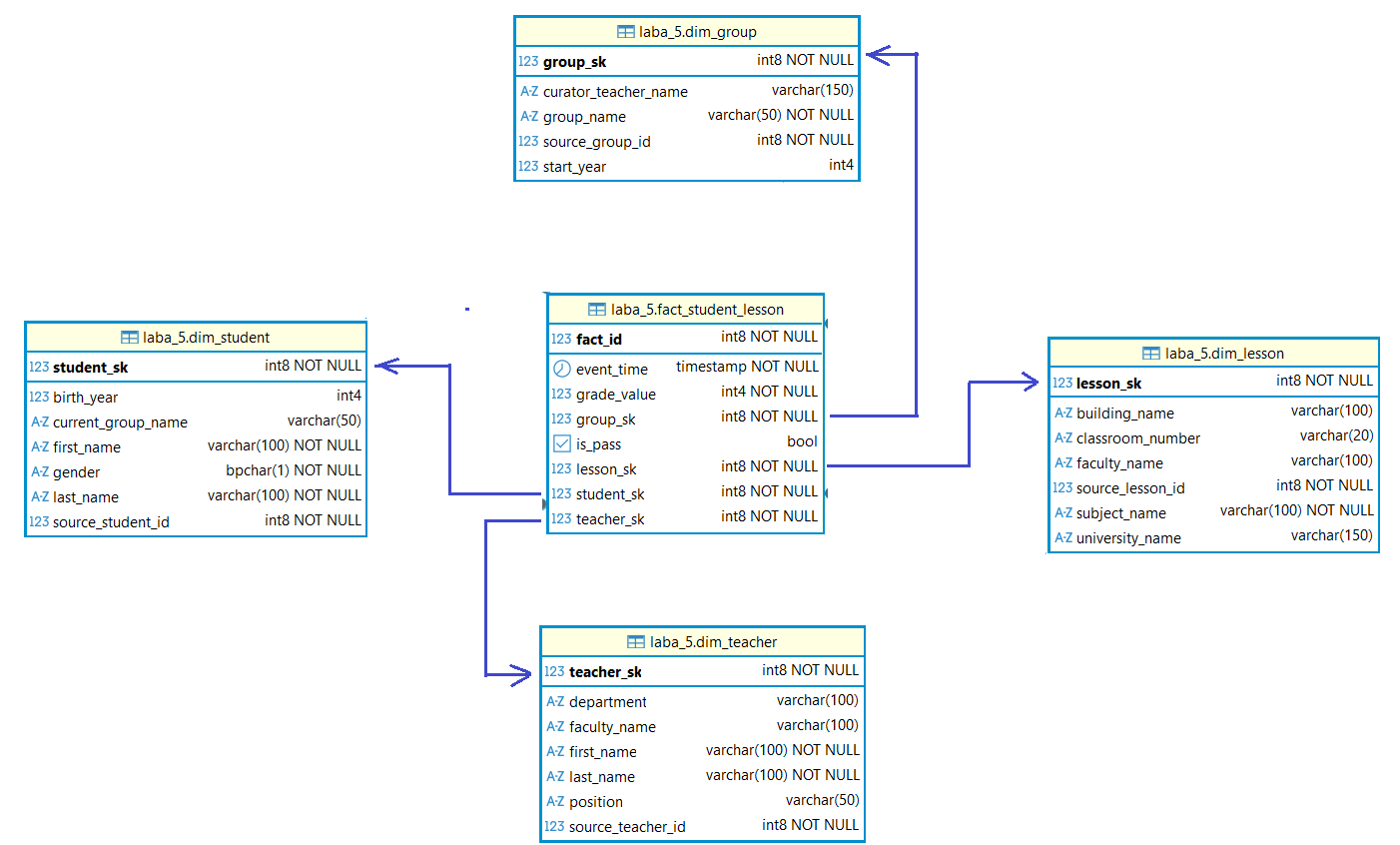
■ is\_pass: BOOLEAN, NULL (сдал/не сдал)

## 5. Физическая модель

Использована **схема «звезда»**:

В центре — таблица фактов fact\_student\_lesson.

Вокруг — измерения dim\_student, dim\_lesson, dim\_teacher, dim\_group.

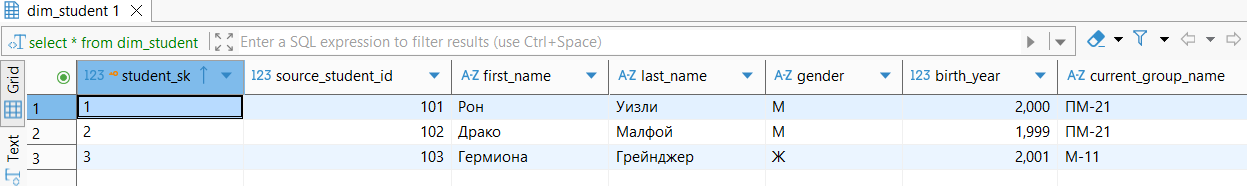


**6.Примеры аналитических запросов**

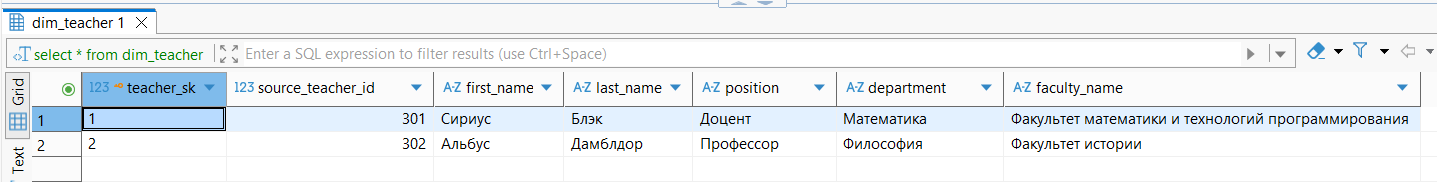
Данный скрипт подублирован в sql-файле homework\_5\_university\_queries.sql

Текущее наполнение таблиц:

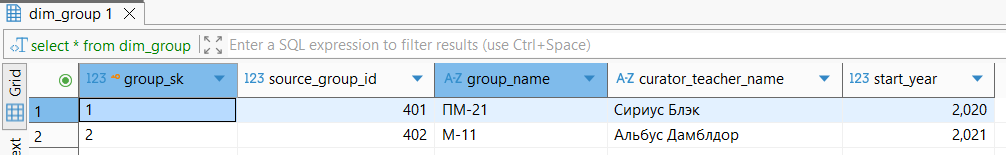
dim\_student



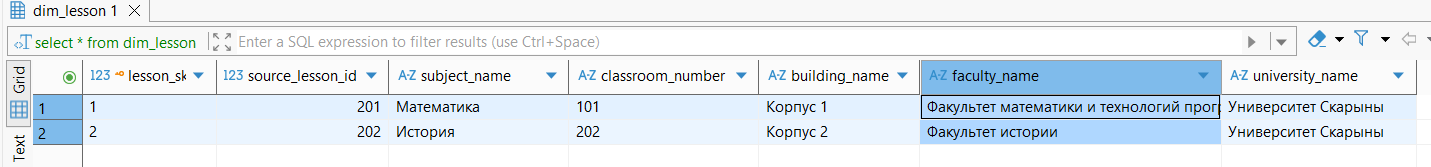
dim\_teacher



dim\_group



dim\_lesson



### fact\_student\_lesson

### 1. Средний балл по предметам - какие предметы в среднем сложнее или проще

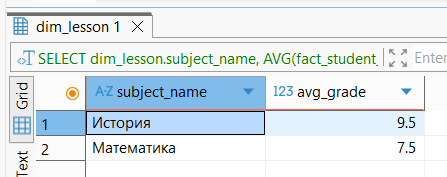
SELECT dim\_lesson.subject\_name, AVG(fact\_student\_lesson.grade\_value) AS avg\_grade

FROM fact\_student\_lesson

JOIN dim\_lesson

ON fact\_student\_lesson.lesson\_sk = dim\_lesson.lesson\_sk

GROUP BY dim\_lesson.subject\_name;



### 2. Количество студентов в каждой группе - сколько студентов учится в каждой группе

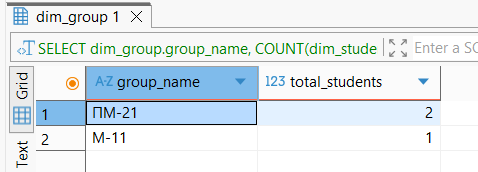
SELECT dim\_group.group\_name, COUNT(dim\_student.student\_sk) AS total\_students

FROM dim\_student

JOIN dim\_group

ON dim\_student.current\_group\_name = dim\_group.group\_name

GROUP BY dim\_group.group\_name;



### 3. Средний балл по группам - какие студенческие группы учатся лучше, а какие хуже

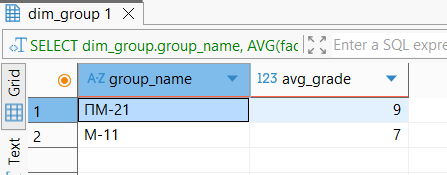
SELECT dim\_group.group\_name, AVG(fact\_student\_lesson.grade\_value) AS avg\_grade

FROM fact\_student\_lesson

JOIN dim\_group

ON fact\_student\_lesson.group\_sk = dim\_group.group\_sk

GROUP BY dim\_group.group\_name;



### 4.Ученики с высоким средним баллом (≥9) - какие студенты в каждой группе учатся на средний балл 9 и выше

SELECT dim\_group.group\_name,

dim\_student.first\_name,

dim\_student.last\_name,

AVG(fact\_student\_lesson.grade\_value) AS avg\_grade

FROM fact\_student\_lesson

JOIN dim\_student

ON fact\_student\_lesson.student\_sk = dim\_student.student\_sk

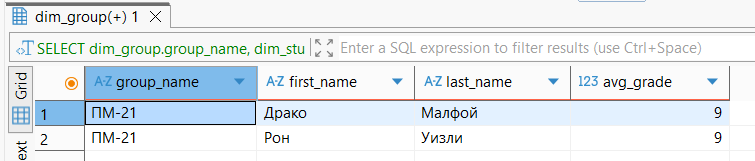
JOIN dim\_group

ON fact\_student\_lesson.group\_sk = dim\_group.group\_sk

GROUP BY dim\_group.group\_name, dim\_student.first\_name, dim\_student.last\_name

HAVING AVG(fact\_student\_lesson.grade\_value) >= 9

ORDER BY dim\_group.group\_name, avg\_grade DESC;



**Вывод:**

В ходе работы было спроектировано **хранилище данных (Data Warehouse)** для университетской базы, ориентированное на анализ успеваемости студентов.

В частности:

- Определён бизнес-процесс: оценивание студентов по занятиям.

- Сформулирован уровень детализации - **одна строка факта соответствует одной оценке студента за конкретное занятие**.

- Разработаны таблицы измерений (студенты, преподаватели, занятия, группы) и таблица фактов (получение оценки студентом за урок).

- Построена физическая модель в виде **схемы «звезда»**, которая является оптимальной для простоты запросов и аналитики.

- Подготовлен набор аналитических SQL-запросов, позволяющих исследовать ряд вопросов.