

Проект по курсу «Базы данных»

Основные правила:

- Проект можно выполнять группой до 4х человек
- Дедлайны по заданиям:
 - по заданиям 1 – 2с: 29 ноября 2023;
 - по остальным заданиям – 13 декабря 2024.
- Максимальный балл за работу ставится при условии своевременного и верного выполнения всех заданий.
- Если студент не приступил к сдаче задания к дедлайну, баллы за него автоматически обнуляются, а задание считается *непринятым*.
- Если студент приступил к сдаче задания до дедлайна, но прислал нерелевантную абракадабру, задание тоже автоматически оценивается 0 баллами и считается *непринятым*.
- Если студент начал сдавать задание, но не все замечания семинариста были исправлены, баллы за задание снижаются. Конкретный балл за задание будет определен семинаристом в зависимости от критичности неисправленных замечаний.
- Проект выполняется итеративно, т.е. нельзя приступать к сдаче следующего задания или пункта задания, пока семинаристом не было принято предыдущее.
- В конце семестра проводится защита проекта.

Цели работы: получение практических навыков работы с промышленными СУБД, проектирование БД (концептуальное, логическое, физическое), создание хранимых процедур, представлений, триггеров, индексов.

Оборудование: персональные ноутбуки с установленной СУБД PostgreSQL.

Задание:

1. Выбрать предметную область для моделирования, согласовать ее с семинаристом. В области должно быть 6-7 основных сущностей, как минимум одна из них – факт (подробнее [тут](#)), остальные – измерения (подробнее [тут](#)).
2. Спроектировать концептуальную, логическую и физическую модели базы данных. Проектировать можно в любом удобном редакторе, но желательно – [draw.io](#). а. Концептуальная модель: все сущности из п.1 с указанием связей в нотации «воронья лапка».

Должно получиться 4–6 основных сущностей.

- b. Логическая модель: описание таблиц и их атрибутного состава, а также указание связей в нотации «воронья лапка». БД должна находиться во 2НФ или в 3НФ (аргументировать свой выбор). Для описания использовать ER-диаграмму в нотации «воронья лапка».

Хотя бы одну таблицу необходимо сделать версионной. На выбор предоставляются SCD2 и SCD4.

После нормализации должно получиться ~8–10 сущностей.

- c. Физическая модель: описание хранения таблиц в СУБД.

По одной таблице на каждый объект из пункта 2b.

Приблизительный формат для описания каждой таблицы:

GOODS Товары				
PK/FK	Название	Описание	Тип данных	Ограничение
PK	GOOD_ID	Идентификатор товара	INTEGER	NOT NULL

	EXTRA_DESC	Доп. описание товара	TEXT	
--	------------	----------------------------	------	--

3. Подготовить DDL скрипты и создать свою базу в СУБД.
Добавить ограничения (CONSTRAINT) по ~10–15 атрибутам и 2 таблицам.
Таблицы должны быть созданы в отдельной схеме.
4. Заполнить БД данными (по 5–10 записей в каждую таблицу). Для заполнения можно использовать как INSERT (минимум по 1 строке в каждую таблицу), так и внешние источники данных (XLS, CSV).
5. Написать CRUD-запросы (INSERT, SELECT, UPDATE, DELETE) к двум таблицам БД.
6. Сформулировать 5 смысловых запросов к БД словами. Написать их на SQL.
Запросы должны содержать (необязательно всё в одном запросе): а. GROUP BY + HAVING;
б. ORDER BY;
в. оконные функции.
Вместе с кодом приложить ваши ожидания от работы запроса.
Пример: «В результате выполнения Запроса 1 будут получены суммарные остатки каждого товара на всех складах».
7. Создать по 1 представлению на каждую таблицу.
Все представления должны храниться в отдельной схеме с представлениями. В представлениях должен быть реализован механизм маскирования персональных данных и скрывания технических полей (суррогатных ключей, полей версионности и т.п.).
Для сокрытия полей с персональными данными недостаточно просто целиком удалить столбец с данными. Например, для номера карты можно использовать маскирование вида: 4276*****0000.
8. Создать 3 сложных представления.
Представления должны соединять несколько таблиц с целью получения осмысленной сводной таблицы, например статистика продаж, частота поставок на склад и т. д.
Вместе с кодом приложить описание представления.
9. Создать 2 триггера на любые таблицы своей БД. Логика работы обговаривается с семинаристом.
10. Создать 2 хранимых процедуры в своей БД. Логика работы обговаривается с семинаристом.
- 11.*** Подключиться к базе через ORM в Python и выполнить CRUD-запросы и смысловые аналитические запросы
12. Выложить результаты выполнения п. 1—11 на Github/Gitlab/что-то подобное.
13. Защитить свой семестровый проект.
Быть готовым отвечать, почему все работает именно так, а не иначе. Быть готовым написать несколько запросов к своей БД в процессе защиты. Быть готовым ответить на некоторые теоретические вопросы. Например, чем 2НФ отличается от 3НФ, какими бывают ключи, найти альтернативные ключи для вашей таблицы и т.д.

Вес заданий:

Задание	Макс. балл	Детализация баллов	Комментарий для сдачи
1	0	-	Необходимо согласовать с семинаристом вне семинаров

2	0	-	Если этот пункт не согласован – за весь проект 0.
3	1	за скрипты без CONSTRAINT – не более 0.5 итоговых балла	Быть готовым написать новые DDL, объяснить логику существующих
4	1	~0.15 за заполнение каждой таблицы	Быть готовым показать, как всё работает, очистить и заново наполнить БД
5	0.5	~0.0675 за запрос ☺	Быть готовым написать новые CRUD-запросы на зачете
6	2.5	0.1 за простые SELECT запросы 0.3 за запросы с группировками и соединениями 0.5 за запросы с версионностью/ оконными функциями	Логика запросов предварительно согласуется с семинаристом. Убедитесь, что запросы выдают то, что вы ожидали. Можно написать больше, чем 5 запросов.
7	1	0.1 за простые представления 0.2 за представления с маскированием	Маскирование: полное, частичное
8	1	~0.33 за представление	Логика представлений согласовывается заранее
9	1.5	~0.75 за триггер	Логика триггеров согласовывается заранее
10	1.5	~0.75 за процедуру	Логика процедуры согласовывается заранее
11	до 2	произвольно	Логику лучше обговорить с семинаристом заранее