

# Databases, Data management

## Course resources



[Elmasri R, Navathe S. Fundamentals of Database Systems, VII-th ed.](#)

[PostgreSQL](#)

[PostgreSQL documentation](#)

[PostgreSQL tutorial](#)

[Planet PostgreSQL](#)

[DBeaver](#)

[PGAdmin 4](#)

[DB-Engines](#)

[Course materials on GitHub](#)

When she was good she was very very good,  
And when she was bad she was horrid.

Henry Longfellow  
(as if written today about database design)

## Учебна програма

### Въведение в базите данни

1. Системи за управление на бази данни. Езикът SQL. DDL и DML. Създаване на бази данни и таблици. Типове данни в SQL.
2. Оператори в SQL. Добавяне, обновяване и премахване на данни. Извличане на данни със SELECT. Филтриране с WHERE.

### Проектиране на релационни БД

1. Свързване на две таблици с JOIN. Видове JOIN - INNER, LEFT, RIGHT и FULL JOIN.
2. JOIN на повече от две таблици. Външни ключове. Действия при UPDATE и DELETE.
3. Нормализиране на бази данни. Първа, втора и трета нормална форма. Денормализация.
4. Естествени и изкуствени ключове. Индекси. Уникален индекс.

### Агрегатни функции, групиране, сортиране

1. Агрегатна функция - дефиниция. Основни агрегатни функции.
2. Филтриране при използване на агрегатни функции - разлики между WHERE и HAVING.
3. Групиране. Агрегатни функции по групи.
4. Сортиране с ORDER BY. Сортиране по няколко колони.
5. Ограничаване на броя записи в резултата. LIMIT и OFFSET. Странициране на резултата.

### Използване на БД от програми

1. Програмни интерфейси и драйвери за достъп до СУБД. Основни компоненти - връзка, транзакция, заявка, параметър.
2. Създаване на връзка към СУБД. Изпълнение на DDL заявка от програма.
3. Изпълнение на DML заявки. "SQL injection" атаки. Параметризирани заявки.
4. Транзакции. ACID (атомарност, поддържане на целостта, изолация и издръжливост).

### Изгледи, обединения

1. Изгледи. Създаване на изгледи. Разлика между изглед и таблица.
2. Обединения с UNION и UNION ALL. Особенности при използване на UNION.

### Функции и процедури

1. Скалярни функции. Създаване на функции. Език за дефиниране на функции и процедури. Блокове BEGIN-END. Детерминистичност на функция. Използване на функции в SELECT заявка.
2. Процедури. Създаване на процедури. Входни, изходни и входо-изходни параметри. Връщане на табличен резултат от процедура чрез временна таблица. Извикване на процедура.

### Тригери

1. Тригер. Видове тригери според извършеното действие – INSERT/UPDATE/DELETE.
2. Промяна на данните в тригер. Видове тригери според времето на изпълнение – BEFORE и AFTER тригери. OLD и NEW. Ред на изпълнение на тригерите.

### Темпорални модели на данни

1. Понятие за темпорален модел. Логическо валидно време и транзакционно време. Моделиране на момент от време и период от време в релационна база данни.
2. Темпорален модел с една дата. Автоматично попълване с тригери. Справки върху актуални данни. Справки върху историята на измененията. Справки за данните към определена дата. Справки върху няколко таблици едновременно.
3. Темпорален модел с две дати. Автоматично попълване с тригери. Справки върху актуални данни. Справки върху историята на измененията. Справки за данните към определена дата. Справки върху няколко таблици едновременно.

№	Наименование на разделите	Минимален брой часове
1.	Задачи за проектиране на релационни БД и боравене с Агрегатни функции, групиране, сортиране	32
2.	Задачи за функции, процедури и тригери	16
3.	Задачи за темпорални данни	16
	<b>Общ минимален брой часове:</b>	<b>64</b>
	<b>Резерв:</b>	<b>8</b>
	<b>Общ брой часове:</b>	<b>72</b>

## Още SQL

- [Таблични](#) и [скаларни](#) под-заявки (subqueries). [Aliasing](#)
- [WITH](#) клауза (Common Table Expressions)
- [OVER](#) клауза (window функции). [DISTINCT \[ON\]](#) клауза
- [Условни изрази](#). Условна агрегация. [FILTER](#) клауза. Агрегатни функции
- [ARRAY](#). Flattening и агрегация.
- [LATERAL JOIN](#)
- [Регулярни изрази](#)
- Дефиниране на [типове](#) и [оператори](#)
- Динамичен SQL. SQL injection
- [MVCC](#)

## Инфраструктура

- ["Чужди" данни](#)
- [Partitioning](#)
- [Репликация](#)
- Оператори [LISTEN/NOTIFY](#)

## NoSQL в RDBMS. JSON и XML

- Защо и кога е подходящо използване на **JSON** и **XML**
- [JSON Оператори и функции](#). [XML Оператори и функции](#)
- Flattening и агрегация
- Индекси
- Динамични възможности
- Отново за връзката с други езици

## Data warehouse, Data lake, Data lakehouse

- Защо и кога е подходящо изграждане на DWH
- Модели на Kimball, Inmon, Data vault. Data lake и Data lakehouse
- ETL и ELT