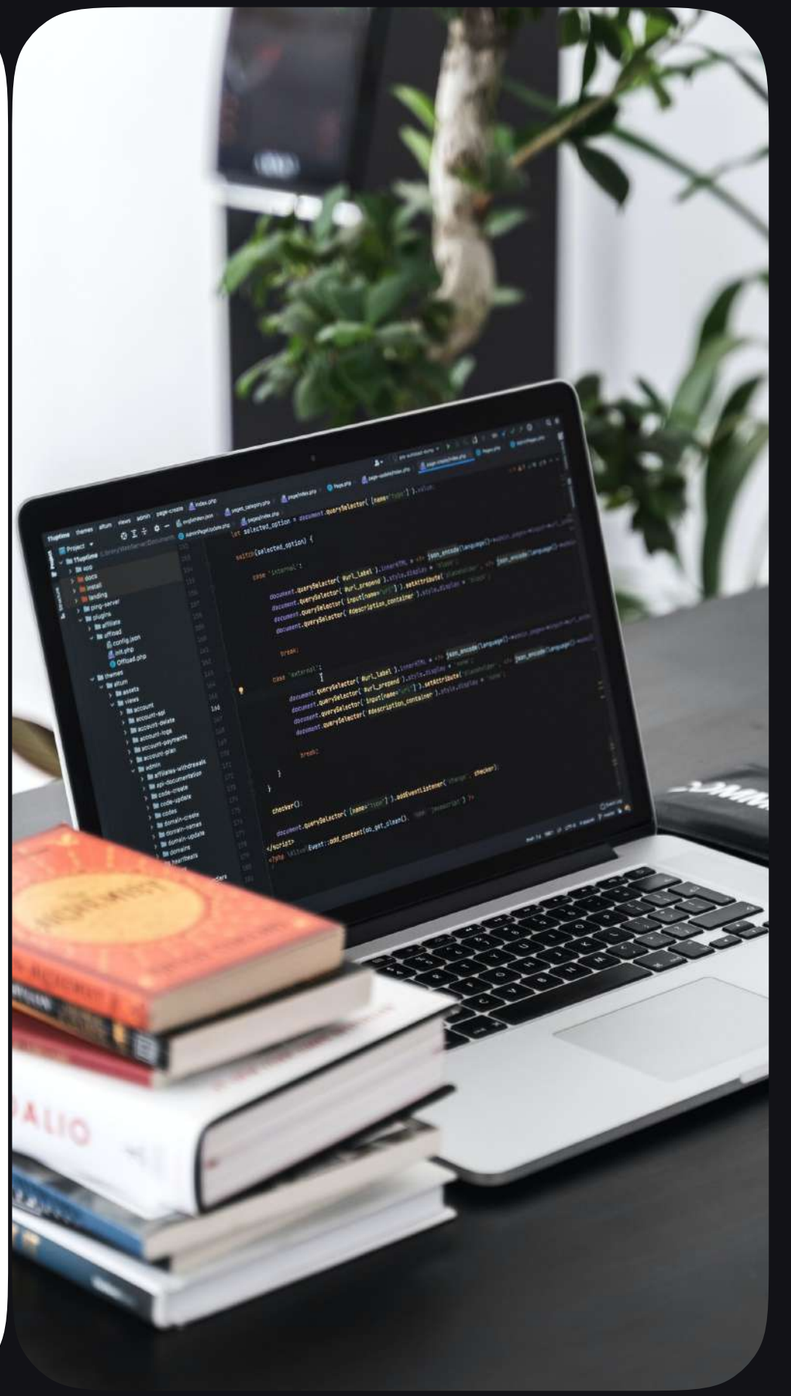
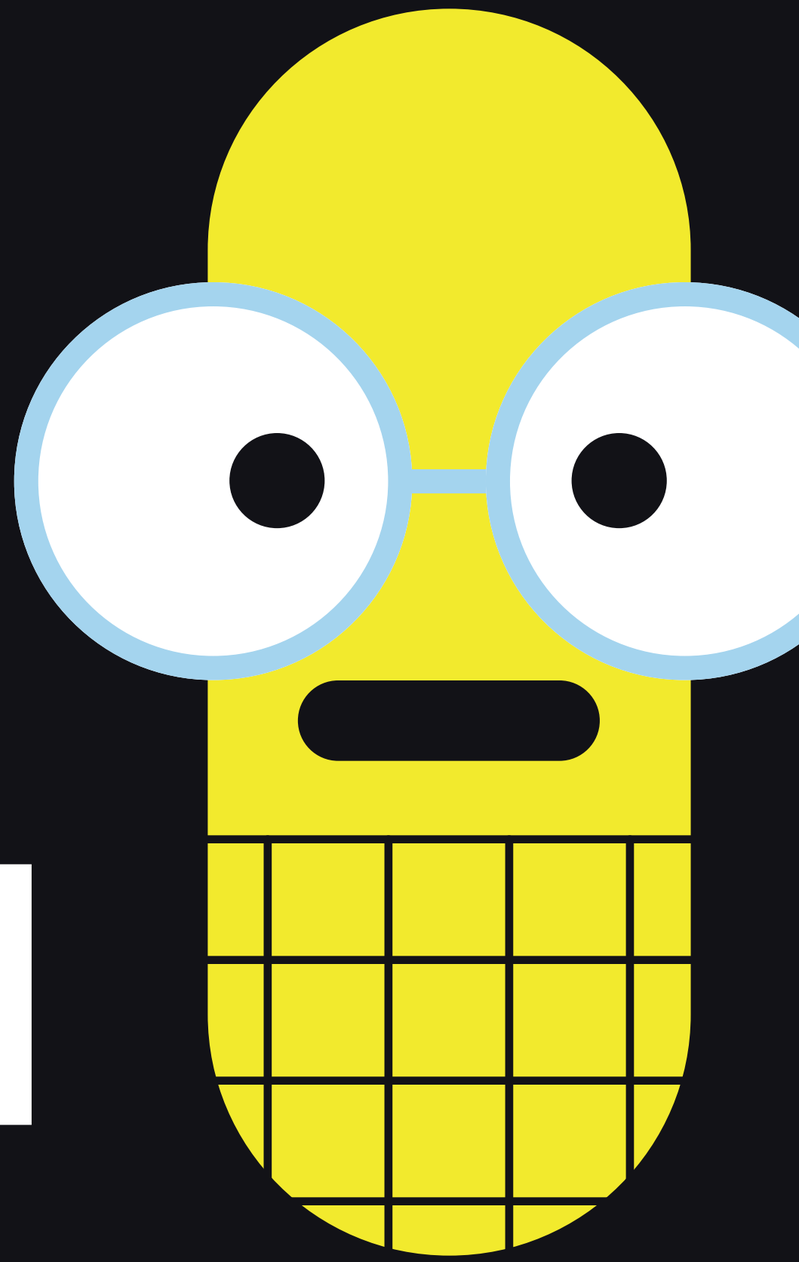
Занятие 7

SQL.
Создание SQL-таблицы



Теория



Базы данных

База данных — централизованное хранилище данных, обеспечивающее хранение, доступ, первичную обработку и поиск информации.

Базы данных разделяются на:

- Иерархические
- Сетевые
- Реляционные
- Объектно-ориентированные

Вся необходимая информация, которая должна храниться в базе данных, размещается в таблицах. Таблица является основным элементом любой реляционной базы данных.

SQL (Structured Query Language)

SQL — представляет из себя структурированный язык запросов, который ориентирован на работу с табличными (реляционными) базами данных.

Система управления базами данных (СУБД)

Для работы с SQL кодом необходима система управления базами данных (СУБД), которая предоставляет функционал для работы с базами данных.

Система управления базами данных (СУБД) — совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями.

Базовые функции СУБД

Интерпретация запросов пользователя, сформированных на специальном языке.

Определение данных (создание и поддержка специальных объектов, хранящих поступающие от пользователя данные, ведение внутреннего реестра объектов и их характеристик — так называемого словаря данных).

Исполнение запросов по выбору, изменению или удалению существующих данных или добавлению новых данных.

Базовые функции СУБД

Безопасность (контроль запросов пользователя на предмет попытки нарушения правил безопасности и целостности, задаваемых при определении данных).

Производительность (поддержка специальных структур для обеспечения максимально быстрого поиска нужных данных).

Архивирование и **восстановление** данных.

Модель данных в реляционных СУБД

По типу модели данных СУБД делятся на:

- **Реляционные**
- **Иерархические**
- **Сетевые**
- **Объектно-ориентированные**
- **Объектно-реляционные**

Реляционная СУБД представляет собой совокупность именованных двумерных таблиц данных, логически связанных (находящихся в отношении) между собой.

Реляционная БД

Таблицы состоят из строк и именованных столбцов:

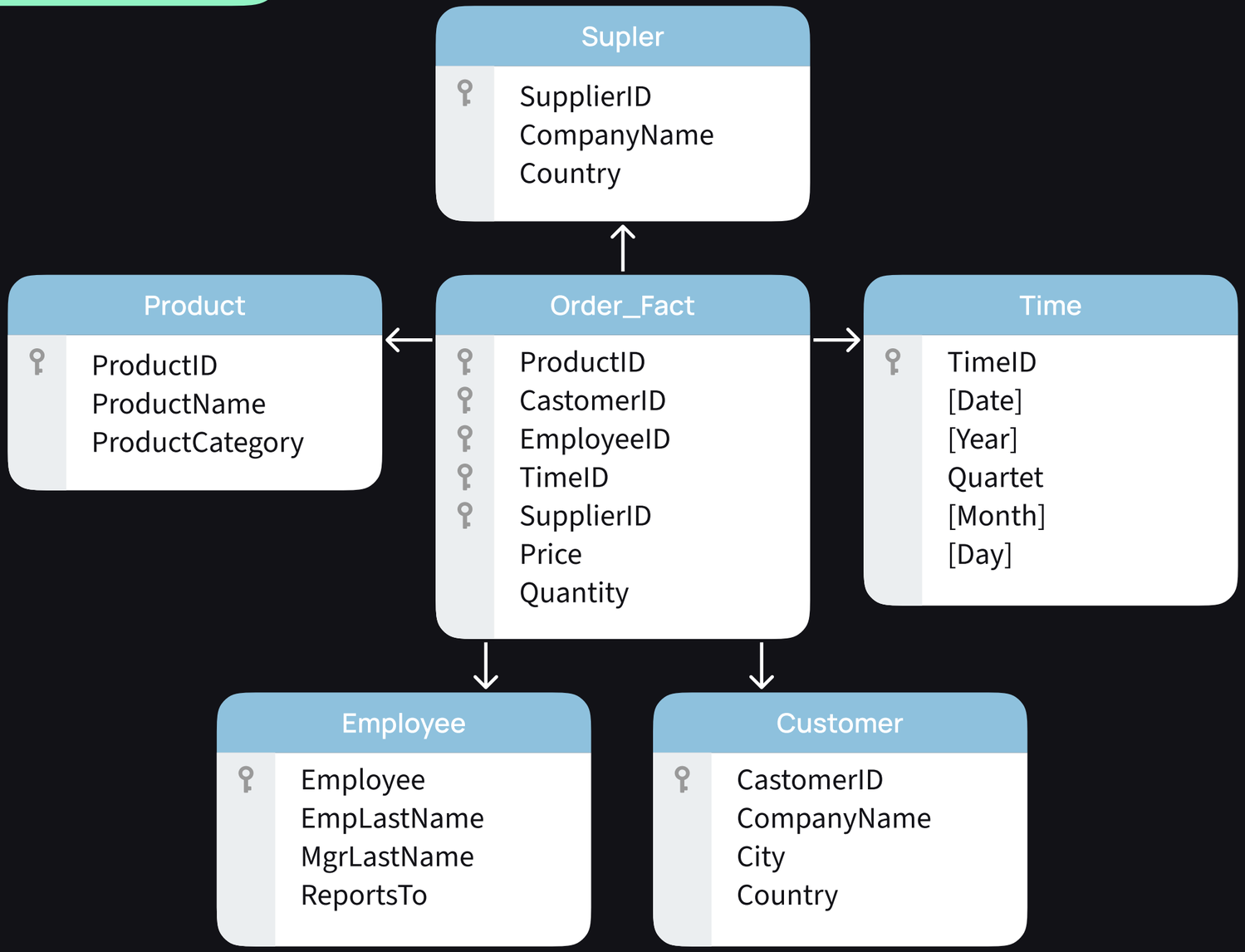
Строки представляют собой экземпляры информационного объекта.

Строки иногда называют **записями,** а столбцы — **полями записи.**

Столбцы — атрибуты объекта.

Таким образом, в реляционной модели все данные представлены для пользователя в виде таблиц значений данных, и все операции над базой сводятся к манипулированию таблицами.

Реляционная БД



Связь в реляционной БД

Связи между отдельными таблицами в реляционной модели в явном виде могут не описываться.

Они устанавливаются пользователем при написании запроса на выборку данных и представляют собой условия равенства значений соответствующих полей.

Связь в реляционной БД

Первичный ключ (главный ключ, primary key, PK). Представляет собой столбец или совокупность столбцов, значения которых однозначно идентифицируют строки.

Вторичный ключ (внешний, foreign key, FK) — столбец или совокупность столбцов, которые в данной таблице не являются первичными ключами, но являются первичными ключами в другой таблице.

Связь в реляционной БД

Сотрудники

Табельный №	Фамилия	Должность	№ отдела
1	Иванов	Начальник	15
2	Петров	Инженер	15
3	Сидоров	Менеджер	10

Отделы

№ отдела	Наименование
15	Производственный отдел
10	Отдел продаж

Ограничения целостности

Целостность базы данных (database integrity) — соответствие имеющейся в базе данных информации её внутренней логике, структуре и всем явно заданным правилам.

Каждое правило, налагающее некоторое ограничение на возможное состояние базы данных, называется **ограничением целостности** (integrity constraint).

Ограничения целостности могут относиться к разным информационным объектам: **атрибутам, кортежам, отношениям, связям** между ними и т.д.

Ограничения целостности

Для полей (атрибутов) используются следующие виды ограничений:

- Тип и формат поля
- Задание диапазона значений
- Задание домена
- Недопустимость пустого поля
- Проверка на уникальность значения какого-либо поля. Ограничение позволяет избежать записей-дубликатов

Ограничения целостности

1 Ограничения таблицы:

✦	PRIMARY KEY	Имя столбца,..	✦	DEFAULT	<Значение по умолчанию>
✦	UNIQUE	Имя столбца,..	✦	REFERENCES	Имя таблицы [(Имя столбца.,..)] [Ссылочная спецификация]
✦	FOREIGN KEY	Имя столбца,..			
✦	CHECK	Предикат	✦	NOT NULL	

2 Ссылочная спецификация:

- ✦ [**ON UPDATE** {CASCADE | SET NULL | SET DEFAULT | RESTRICTED| NO ACTION}]
- ✦ [**ON DELETE** {CASCADE | SET NULL | SET DEFAULT | RESTRICTED| NO ACTION}]

Нормализация

Основная цель **нормализации** — устранение избыточности данных.

- ✓ Первая нормальная форма (**1НФ, 1NF**)
- ✓ Вторая нормальная форма (**2НФ, 2NF**)
- ✓ Третья нормальная форма (**3НФ, 3NF**)
- ✓ Нормальная форма Бойса-Кодда (**НФБК, BCNF**)
- ✓ Четвёртая нормальная форма (**4НФ, 4NF**)
- ✓ Пятая нормальная форма (**5НФ, 5NF**)
- ✓ Доменно-ключевая нормальная форма (**ДКНФ, DKNF**)

Нормализация модели данных

Первая нормальная форма:

информация в каждом поле таблицы является неделимой и не может быть разбита на подгруппы.

...	Иванов, 15 отдел, начальник	...

Фамилия	Должность	№ отдела
Иванов	Начальник	15

Нормализация модели данных

Вторая нормальная форма:

таблица соответствует 1НФ и в таблице нет неключевых атрибутов, зависящих от части сложного (состоящего из нескольких столбцов) первичного ключа.

№ отдела	Должность	Отдел	Количество сотрудников
15	Начальник	Производственный отдел	1
15	Инженер	Производственный отдел	5
15	Начальник	Отдел продаж	1
10	Инженер	Отдел продаж	10

Структура

№ отдела	Должность	Количество сотрудников
15	Начальник	1
15	Инженер	5
15	Начальник	1
10	Инженер	10

Отделы

№ отдела	Наименование
15	Производственный отдел
10	Отдел продаж

Нормализация модели данных

Третья нормальная форма:

таблица соответствует первым двум НФ и все неключевые атрибуты зависят только от первичного ключа и не зависят друг от друга.

Сотрудники

Табельный №	Фамилия	Оклад	Наименование отдела	№ отдела
1	Иванов	500	Производственный отдел	15
2	Петров	400	Производственный отдел	15
3	Иванов	600	10	10

Сотрудники

Табельный №	Фамилия	Оклад	№ отдела
1	Иванов	500	15
2	Петров	400	15
3	Иванов	600	10

Отделы

№ отдела	Наименование
15	Производственный отдел
10	Отдел продаж

Язык SQL

SQL (Structured Query Language) — непроцедурный язык взаимодействия приложений и пользователей с реляционными СУБД, состоящий из набора стандартных команд на английском языке.

Отдельные команды изначально никак логически не связаны друг с другом.

Язык SQL

- SQL может использоваться как интерактивный (для выполнения запросов) и как встроенный (для построения прикладных программ).
- Базовый вариант SQL содержит порядка **40** команд (часто еще называемых **запросами** или **операторами**) для выполнения различных действий внутри СУБД.

Операторы SQL

Выделяют следующие группы операторов SQL

- операторы определения объектов базы данных (Data Definition Language — **DDL)**
- операторы манипулирования данными (Data Manipulation Language — **DML**)
- команды управления транзакциями (Transaction Control Language — **TCL**)
- операторы защиты и управления данными (Data Control Language — **DCL**)

Операторы SQL

Операторы DDL — определения объектов базы данных :

CREATE DATABASE	создать базу данных	CREATE DOMAIN	создать домен
DROP DATABASE	удалить базы данных	ALTER DOMAIN	изменить домен
CREATE TABLE	создать таблицу	DROP DOMAIN	удалить домен
ALTER TABLE	изменить таблицу	CREATE VIEW	создать представление
DROP TABLE	удалить таблицу	DROP VIEW	удалить представление

Операторы SQL

Операторы DML — манипулирования данными:

SELECT	отобрать строки из таблиц
INSERT	добавить строки в таблицу
UPDATE	изменить строки в таблице
DELETE	удалить строки в таблице

Операторы SQL

Команды управления транзакциями TCL

Используются для управления изменениями данных, производимыми DML-командами. С их помощью несколько DML-команд могут быть объединены в единое логическое целое, называемое транзакцией.

COMMIT	завершить транзакцию и зафиксировать все изменения в БД
ROLLBACK	отменить транзакцию и отменить все изменения в БД
SET TRANSACTION	установить некоторые условия выполнения транзакции

Операторы SQL

Операторы защиты и управления данными — DCL

GRANT

предоставить привилегии пользователю или приложению на манипулирование объектами

REVOKE

отменить привилегии пользователя или приложения

Язык SQL

звездочка (*)	для обозначения «все»
квадратные скобки ([])	конструкции, заключенные в эти скобки, являются необязательными (т.е. могут быть опущены)
фигурные скобки ({})	конструкции, заключенные в эти скобки, должны рассматриваться как целые синтаксические единицы
Многоточие (...)	указывает на то, что непосредственно предшествующая ему синтаксическая единица факультативно может повторяться один или более раз
прямая черта (|)	означает наличие выбора из двух или более возможностей

Язык SQL

точка с запятой (;)	завершающий элемент предложений SQL
запятая (,)	используется для разделения элементов списков
пробелы ( )	могут вводиться для повышения наглядности между любыми синтаксическими конструкциями предложений SQL
прописные жирные латинские буквы и символы	используются для написания конструкций языка SQL
строчные буквы	используются для написания конструкций, которые должны заменяться конкретными значениями, выбранными пользователем

SELECT

Для выборки данных используется команда **SELECT**

SELECT [**DISTINCT**] <список столбцов>

FROM <имя таблицы> [**JOIN** <имя таблицы> **ON** <условия связывания>]

[**WHERE** <условия выборки>]

[**GROUP BY** <список столбцов для группировки> [**HAVING** <условия выборки групп>]]

[**ORDER BY** <список столбцов для сортировки>]

Секция DISTINCT

Если в результирующем наборе данных встречаются одинаковые строки (значения всех полей совпадают), можно от них избавиться, указав ключевое слово **DISTINCT** перед списком столбцов.

```
SELECT DISTINCT Position FROM Employees
```

Секция FROM

Перечень таблиц, из которых производится выборка данных, указывается в секции **FROM**. Выборка возможна как из одной таблицы, так и из нескольких логически взаимосвязанных.

- Логическая взаимосвязь осуществляется с помощью подсекции **JOIN**.
- На каждую логическую связь пишется отдельная подсекция.
- Внутри подсекции указывается условие связи двух таблиц (обычно по условию равенства первичных и вторичных ключей).