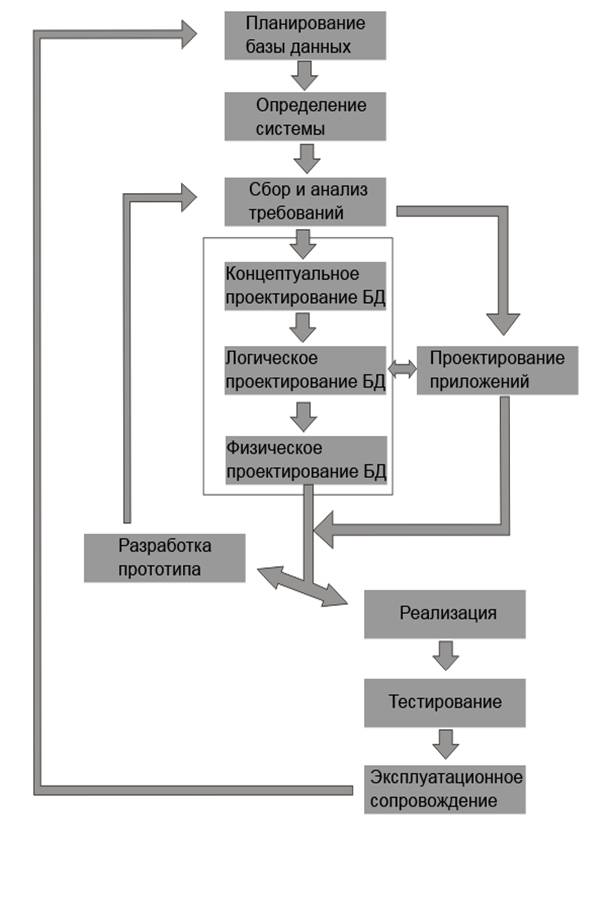
Глава 1

Лекция 3

Самые важные этапы проектирования:

1. Планирование
2. Сбор и анализ информации
3. Проектирование
4. Оценка БД



СУБД

Из чего состоит комп. СУБД:

1. Ядро – отвечает за управление данными, состоит из
   1. Менеджера данных
   2. Менеджер буферов
   3. М. транзакций (Транзакция — это логическая единица работы с базой данных, которая объединяет несколько операций (например, INSERT, UPDATE, DELETE) в одно целое. Главная идея: либо выполняются все операции транзакции, либо не выполняется ни одна.)
   4. М. журнала изменений
   5. Другие
2. Подсистема поддержки времени выполнения запросов
3. Набор утилит

Примеры СУБД

1. MS SQL Server
2. Oracle DB
3. PostgreSQL
4. MySQL
5. MS Access

Модели Данных (ДХ) (МД)

**Предметная область** – Чётко очерченная часть реального мира, данные, которых должны отразить в БД

**Модель ДХ** – способ описания/структурирования и логической организации дх

Виды МД:

1. Иерархическая МД – форма: дерево с жесткими вертикальными связями
2. Сетевая МД – как иерархическая, но с горизонтальными связями
3. ОО~~П~~ МД
4. Реляционные (Relation) МД –

ДЗ: 1. Печатать файлы 2. Опрос 3. Прочитать

Лекция 4

Теория БД

**Свойства отношений**

1. Картежи отношений не упорядочена
2. Атрибуты отношений не упорядочены
3. Операция модификация ДХ проходит **построчно**
4. В отношении не должно быть картежей дубликатов (повторяющиеся строки)
5. Значения атомарны

**Первичный ключ** (Primary Key) (PK) – атрибут, значение которого позволяют однозначно идентифицировать каждую строку в таблицу.

**Свойство PK**:

1. PK всегда есть в отношении
2. Всегда один
3. Значение PK уникальны
4. PK по умолчанию не допускает «значение» null
5. На РК по умолчанию создаётся кластеризованный индекс

Простые РК из одного атрибута (аттр), составные из нескольких

РК – естественные (объявляется на аттр существующем в предметной области) и суррогатные (тот, который мы создаём и назначаем самостоятельная)

Чаще всего используем суррогатные простые ключи

**Внешний ключ** (Foreign Key) – столбец для принудительного связывания ДХ меж разными таблицами

**Свойства**: 1. Значениями FK только те значения связанные с ним в РК

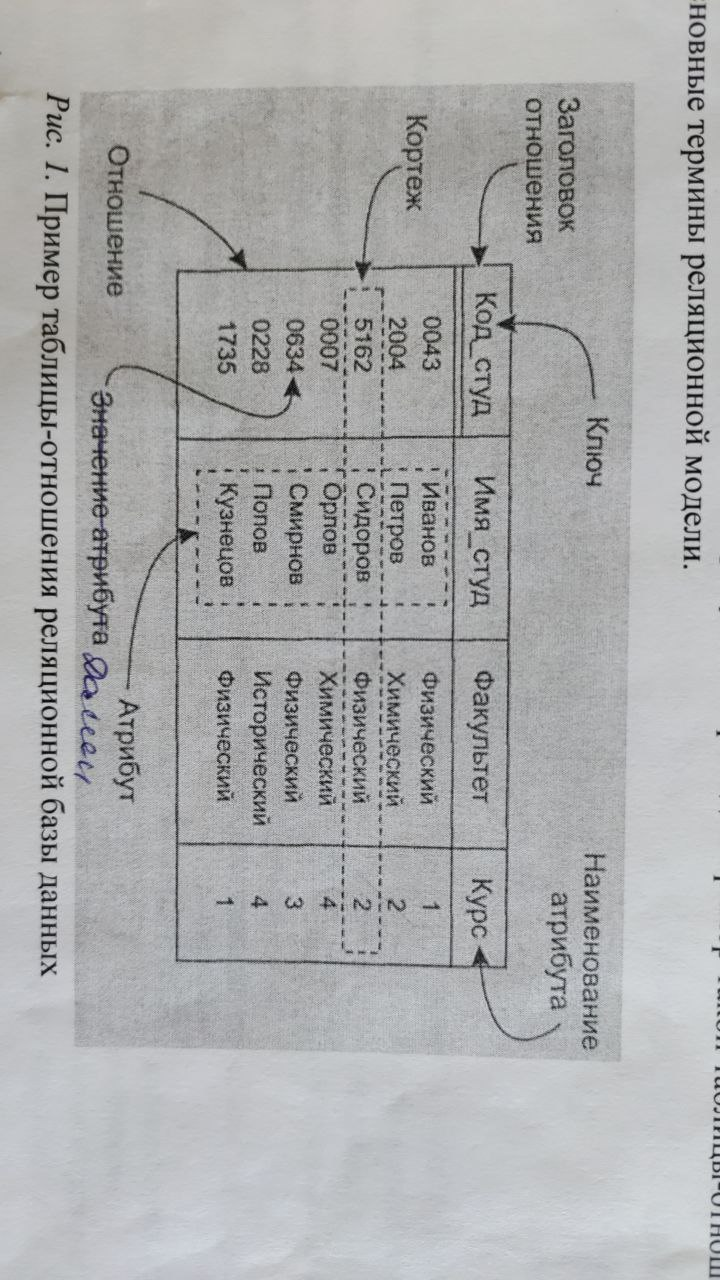
**Первая нормальная форма**

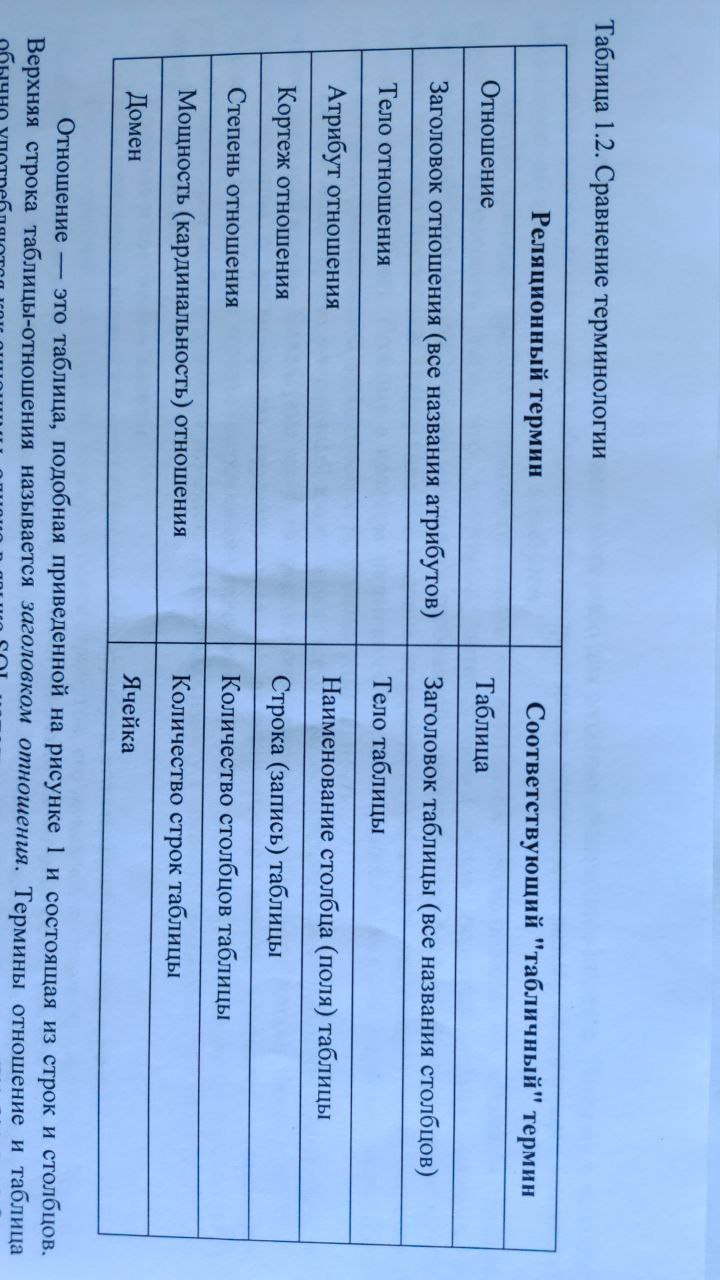
**Первая нормальная форма (1НФ)** — это обычное отношение. Согласно нашему определению отношений, любое отношение автоматически уже находится в 1НФ. Напомним кратко свойства отношений (это и будут свойства 1НФ):

Для того, чтобы устранить зависимость атрибутов от части сложного ключа, нужно произвести **декомпозицию** отношения на несколько отношений. При этом те атрибуты, которые зависят от части сложного ключа, выносятся в отдельное отношение.

Отношение R находится во второй нормальной форме (**2НФ**) тогда и только тогда, когда отношение находится в 1НФ и нет неключевых атрибутов, зависящих от части сложного ключа. (**Неключевой атрибут** — это атрибут, не входящий в состав никакого потенциального ключа).

Атрибуты называются взаимно независимыми, если ни один из них не является функционально зависимым от другого. Отношение находится в **третьей нормальной форме (3НФ)** тогда и только тогда, когда отношение находится в 2НФ и все неключевые атрибуты взаимно независимы.





Лекция 5

file:///D:/Telegram/Типы данных1.pdf

Ограничения

Ограничения (Constraints) – элементарные проверки/условие, выполняются для операции модификации ДХ, если проверка не выполняется, то операция откатывается. К ограничениям относим:

1. РK
2. FK
3. Not Null (NN) (аттр не может иметь «значение» null)
4. Default – задаёт для аттр значение по умолчанию, которое пишется сразу после ограничения
5. Unique (юник) – все значения в аттр должны быть уникальными

Виды типов ДХ (ТДХ): числовые, текстовые, календарные и др.

Примеры в методе:

ТДХ – ограничения аттр, при назначении ТДХ мы руководствуемся принципом «минимальный необходимость»

“Значение” null – null, указатель на отсутствие значения в ячейке, **null – не значение/не тип ДХ.**

Ограничения целостности ДХ (ЦДХ) (О-ДХ):

1. Средство обеспечения доменной целостности – нужны для недопущения ввода в ячейки БД недопустимых значений (через ограничение и ТДХ)
2. Сущностная целостность – не допускает двукратного попадания ДХ об одной сущности в базу. (обесчивается: 1. Свойством unique, 2. …)
3. Ссылочная цел-ость – обеспечивает единство и связность ДХ (Комбинация РК и FK)
   1. Каскадная – определяет действие, предпринимаемое при попытке удалить/обновить РК, на который указывают внешние ключи



Индекс (Key)

Кеу - Физическая струк-ра, которая СУБД использ. для обработки ДХ, они позволяют искать ДХ-е. Создаются автоматически

Лекция 6

Нормализация отношений

Проектирование БД

МДХ – графическое представление, части реального мира (объектов, событий, прочее и связей меж ними), позволяющее проектир-икам и польз-лям БД иметь однозначное представление о её строении. Цель построения МДХ – представление ДХ в понятном виде

Концептуальное проек-оание (прк-ие) – построение информационной модели предметной области, наиболее высокого уровня абстракции

ER – entity relationship

Нотация – набор граф изобр, применимой к той/иной области знаний или деятельности

Нотации ER модели

Сущность – ▭ – класс однотипных объектов, информация о которых будет храниться в БД

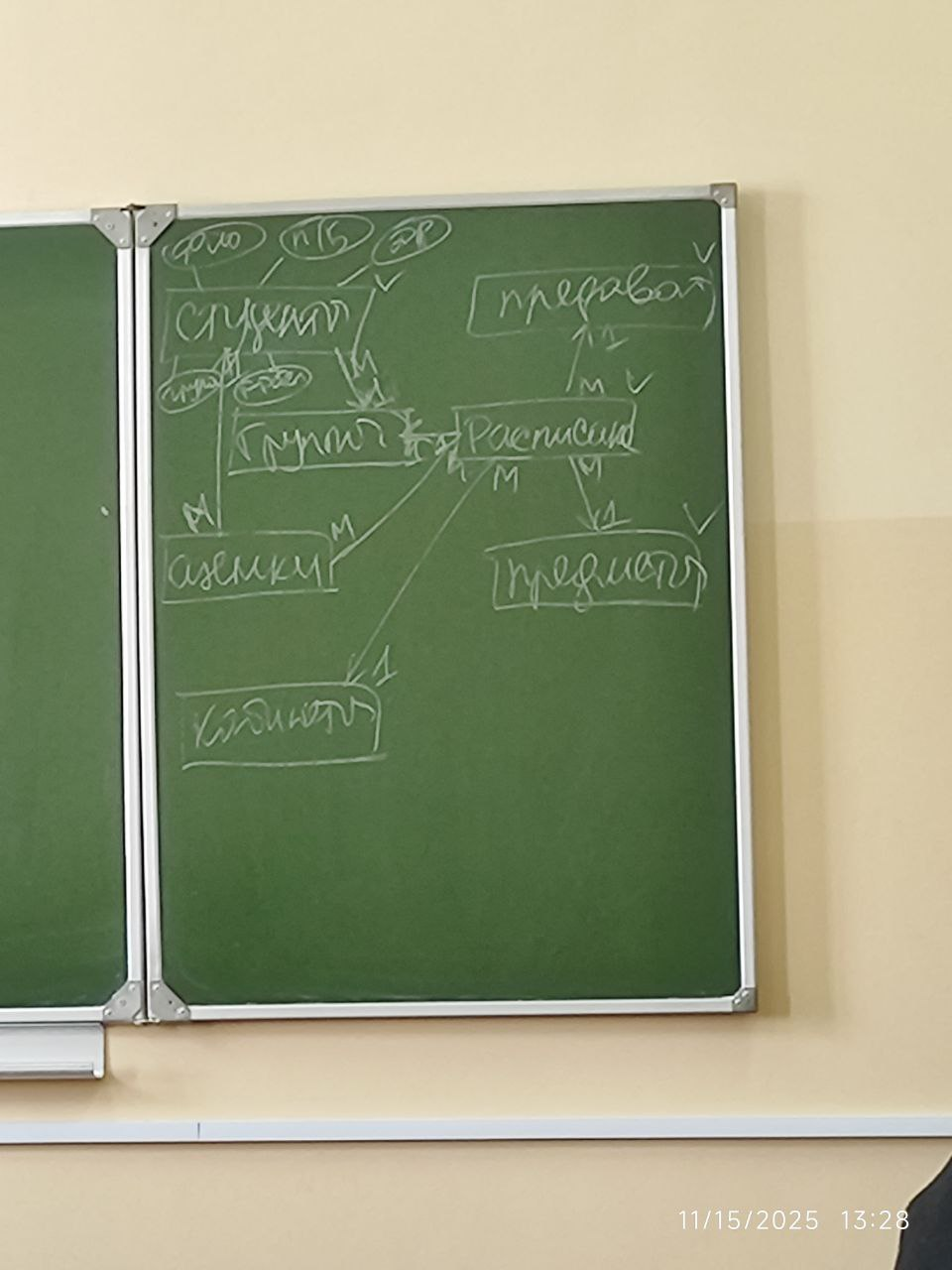
Аттр сущности – ◯ (овал) – любая характеристика сущности для её описания

Связь – ◇→ (→) – Графически изображаемая ассоциация меж сущностями

Очень мало случаев применения многие ко многим

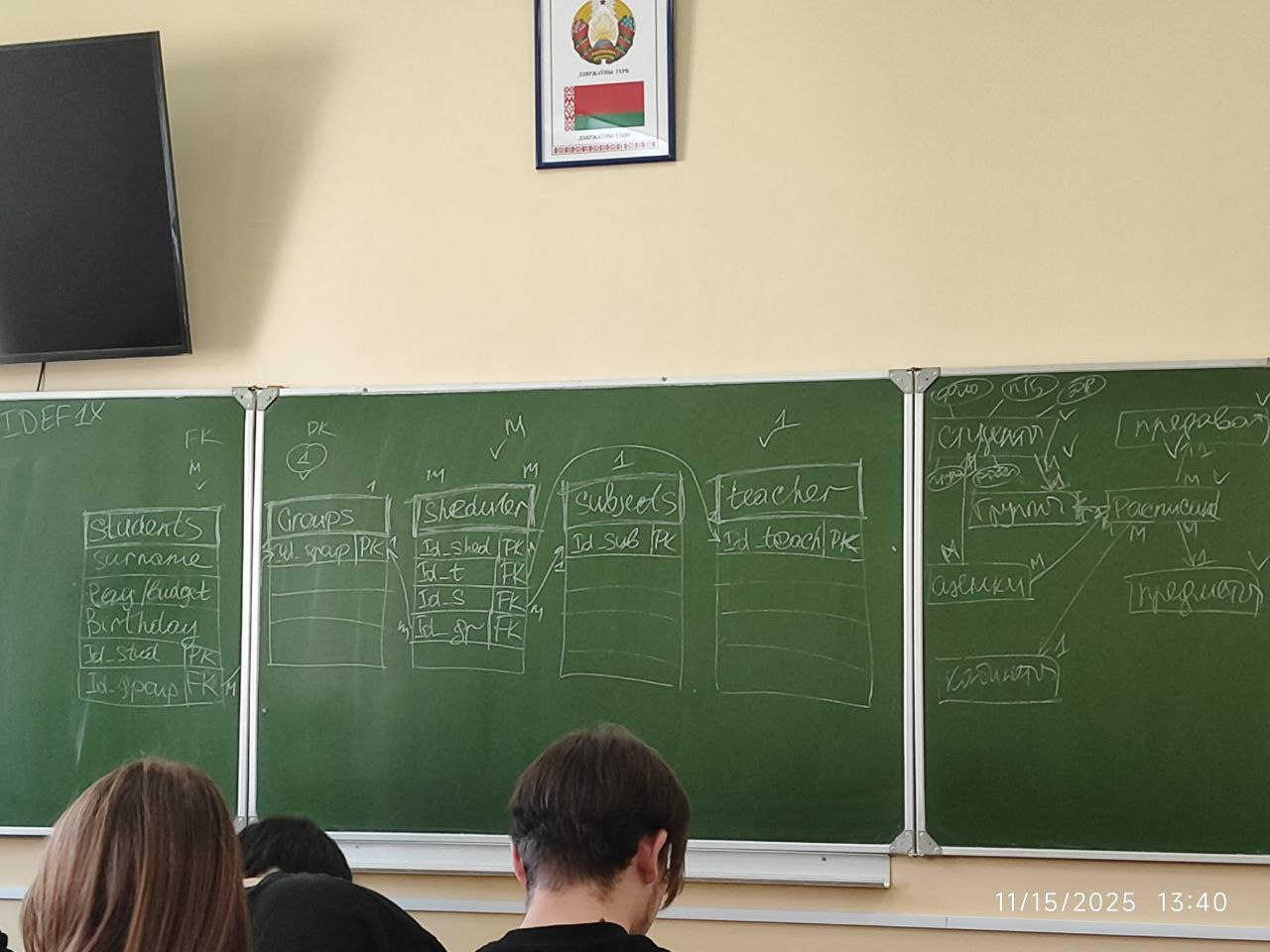
Полная аттр модель – нотация **EFE1X** – даёт детальное представление стр-ры ДХ.

Проц-сс преобразование er модели в 1ef1x: 1. Каждая сущн становится (/) отношением, каждый аттр – аттр отношения, связи превращаются в ключи отношений



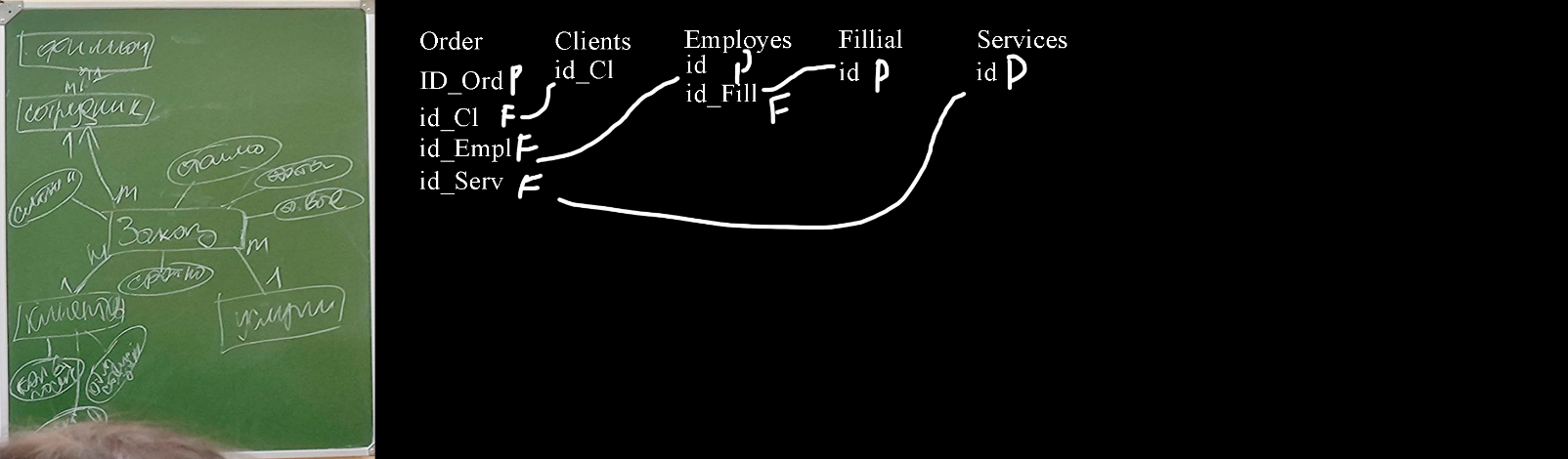
Составляется на любом языке er

1ef1x – на английском



Физическое проектирование – на этом этапе каждое отношение => в таблицу, где подробно описаны характеристики всех аттр, вкл в себя:

1. Название аттр на англ
2. Пояснение, что должно храниться в аттр понятное заказчику
3. ТДХ и ограничения по ТДХ
4. Ограничение аттр



Лекция7

Общее описание языка (стандарта) SQL. Правила составления запроса

1. Запросы начинаются с ключевого слога-глагола
   1. CREATE
   2. DELETE
   3. INSERT
2. Каждый запрос заканчивается знаком **;** называют его разделителем запроса
3. Запросы пишутся на английском языке, но данные можно с любого языка использовать
4. Запросы из нескольких предложений, имеют фиксированный порядок их расположения (Update)
5. Запрос может быть записан в свободной форме и состоять из любого кол-ва строк (не Python, а C#)
6. Для комментирования практически любой вид комментария
7. Из знаков препинания в телезапросов используется только запятые (,) и только для разделения однородных элементов запроса – аттр, отношений (столбцов, таблиц)
8. Название таблиц должны быть уникальными в 1 БД, название столбцов должны быть уникальным в рамках 1 таблицы
9. Буквы латиницы, цифры и нижние подчёркивание. Первые символы должны быть буквы
10. SQL регистронезависимый язык

Лекция 8

DDL

3 оператора ДДЛ:

1. CREATE
2. DROP
3. ALTER

Create database firma;

Стр-ра запроса ДДЛ:

1. Действие (Create, Drop, Acter)
2. Объект БД, к которому применяется запрос
3. На 3 месте имя собственное
4. Дальше некие характеристики запроса

Create table tbl\_name

(clmn\_name type [constraimt],

clmn\_name1 type [constraimt]

Priarmory\_key(clmn\_name),

[ForeignKey(clmn\_name1)

References tbl\_name1(clmn\_names)]);

Create table employers;

(id\_ employee INT AUTO\_INCREMENT,

Surname VARCHAR(20),

id\_depart TINYINT

salary\_hour TINYINT

date\_start\_emp DATE NOT NULL

Primary\_key (id\_ employee),

ForeignKey (id\_depart)

Refrences department(id\_depart));

Вставка данных

Запрос Insert

Insert into tbl\_name

(clm\_name1, clm\_name2, …)

Values

(‘Знач1’, ‘Знач2’…),

(‘Знач1’, ‘Знач2’…);

Insert into employees

(surname, salary\_hour, id\_depart, date, )

Values

(‘Иванов’, 15, 1, curdate()),

(‘Сидоров’, 10, 1, ‘2025-06-01’);

Правила/ограничения оператора Insert

1. Строковые ТДХ вписываются в вертикальных одинарных кавычках. Целочисленные – без кавычек
2. Кол-во столбцов – кол-во данных
3. Если значение аттр – неизвестно и ограничения аттр – позволяют, то вместо значения пишем Default

ДЗ: посмотреть видео, взять МД и сделать их

Минимум 8 кортежей

Лекция 9

Обработка ДХ DQL

Select

Синтаксис

|  |  |
| --- | --- |
| **Запрос** | **Описание** |
| Select | «Выбрать» в нём мы прописываем какие аттр будут выведены из запроса |
| FROM | Прописываем с какими таблицами будет работать нащ запрос |
| Where | Условие выборки ДХ или фильтрация ДХ |
| Group By | Группировка ДХ по указанному аттр  Аттр в Group By и Select должны быть одинаковыми |
| Having | Фильтрация на уровне сгрупированных ДХ |
| Order By | Отвечает за сортировку выборки по аттр (по умолчанию возрастание, если наоборот, то пишем Desc) |
| Limit | Ограничивает кол-во строк выборки  Limit N, M;  С N по M строки мы выведем  Limit M;  С начала и до M |

Задача: вывести все фамилии сотрудников

Select Surname

FROM Employees;

Select \* – выводит всё

Задача вывести сотрудников с зарплатой больше 50

Select \*

FROM Employees

Where Salary\_hour > 50;

Шаблон: **Название аттр оператор сравнения значение**

Задача: вывести ID сотрудников, у которых есть задача

|  |
| --- |
| Select id\_empl  FROM tasks  Group By id\_empl; |

Задача: вывести ID сотрудников у кого больше 4 задач

|  |
| --- |
| Select id\_empl  FROM Task  Group By id\_empl  Having COUNT(id\_empl) > 4; |

Задача: Вывести Ф сотрудников в обратном алфавитном порядке

|  |
| --- |
| Select Surname  FROM Employees  Order By Surname desc; |

Операторы предложений условия

1. Математические: >, <, =
2. Объединение условий: AND, OR

Задача: Вывести сотрудников первого отдела с зарплатой 10 и 15

Select \*

FROM Employees

Where (Salary\_hour = 10 OR Salary\_hour = 15) AND id\_depart = 1;

BETWEEN … AND … Диапазон значений

Задача: Вывести сотрудников, которые зарабатывают от 10 до 20

Select \*

FROM Employees

Where Salary\_hour Between 10 AND 20

Выборка по пустым значениям

IS NULL IS NOT NULL

Задача: вывести данные обо всех завершённых задачах

Select \*

FROM Tasks

Where date\_turn IS NOT NULL;

Оператор объединения условий OR IN

Задача: вывести сотруд. с зарплатой: 10, 15, 20, 25

Select \*

FROM Employees

Where Salary\_hour = IN(10, 15, 20, 25)

ДЗ

#1

Select \*

From Employees

Order by surname, id\_depart;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#2

Select\*

From Employees

Order by Rand()

Limit 1;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#3

Select \*

From Employees

Where (id\_depart between 1 and 3) and (salary\_hour = 15);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#4

SELECT \*

FROM Employees

ORDER BY salary\_hour DESC

LIMIT 1;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#5

Select\*

From Employees

Where (surname like 'А%') and (surname like '%й');

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#6

Select \*

From Tasks

Group By id\_empl

Having Count(id\_task) > 4;

Надо:

SELECT id\_empl, COUNT(id\_task) AS task\_count

FROM Tasks

GROUP BY id\_empl

HAVING COUNT(id\_task) > 4;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#7

Select id\_depart

From Employees

Group By id\_depart

Having Count(id\_employee) > 4;

Надо:

SELECT id\_depart, COUNT(id\_employee) AS emp\_count

FROM Employees

GROUP BY id\_depart

HAVING COUNT(id\_employee) > 4;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#8

Select \*

From Tasks

Where (id\_project IN(1,4,7)) AND (date\_turn IS NOT NULL)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Лекция 10

Первый From, второй Where, Group By, Having, Select, Order By, Limit

Not

В MySQL нет «!=», вместо него есть NOT

Задача. Вывести всех сотрудников, кроме первого

Select \*

From Employees

Where NOT id\_depart = 1;

Оператор выборки по частичному совпадению/по маске

Like

3 метасимвол

«?/\_» – заменяет 1 символ в позиции

«\*/%» - любое кол-во симв в позиции

«[]» – заменяют 1 символ в позиции, внутри можно прописать несколько символов, которые сработают, через оператор OR

З Вывести Ф, которые начинаются на А, Д, Е; заканчиваются на Й, есть Р

SELECT surname

FROM employees

WHERE (

surname LIKE 'А%й'

OR surname LIKE 'Д%й'

OR surname LIKE 'Е%й'

)

AND surname LIKE '%Р%';

**NOW, CURDATA, MOUNTH, YERS, DATE\_ADD, DATE\_DIFF, Форматирование дат и времени, CONCAT, LEFT/RUGHT SUBSTRING, TRIM**

Агрегатные f

MIN(…) –

MAX(…) – максимальное значение

AVG(…) – средние

SUM(…) – сумма

COUNT(…) – кол-во строк

Ограничение:

1. Запрещено в Where, FROM, LIMIT, GROUP BY

JOIN

FROM TAB\_1 JOIN TAB\_2 ON

TAB\_1.CLMN\_1 = TAB\_2.CLMN\_1

FROM department join employees on

department.id\_depart = employees.id\_depart

Вывести Ф сотрудников с задачами в работе

Select distinct surname

From Employees AS empl join task AS t ON

Where t.date\_turn is NULL

Лекция 11

Вложенные селекты (подзапросы) (ВС)

ВС –

Вывести список задач сотрудника Иванова

Select \*

From Tasks

Where id\_empl = (

Select id\_empl

From Employees

Where surname = ‘Иванов’

);

**Ограничение ВС**

1. Всегда контролируем, что возвращает подзапрос и соответствует ли результат его работы месту, где он прописан:
   1. Если подзапрос use в условии как значение, то может возвращать строго 1 столбец
   2. Если возвращает несколько значений, то могут использоваться операторы с множествами значений (IN)
2. Во ВС запрещено использовать Order By
3. ВС нельзя использовать как значение в операторе Between
4. ВС не используются как аргумент в агрегатных функциях
5. Используется
   1. В селекте в кач-ве аттр
   2. Во from в кач-ве таблицы
   3. В where и having в кач-ве условия

Если зарплата сотрудника < 10, «Требуем повышения», иначе «Нормально»

SELECT surname, salary\_hour,

IF(salary\_hour < 10, 'Требуем повышения', 'Нормально') AS status

FROM employees;

Выбрать Ф сотрудников у кого нет заданий

Select surname

From Employees

Where id\_employer IN (

Select distinct id\_empl

From Tasks

Where NOT date\_turn IS NULL

);

Задания по проекту Космос и ещё не выполнены

Select \*

From Tasks

Where date\_turn IS NULL AND id\_project in (

Select id\_project

From Projects

Where project\_name = ‘Космос’

);

Найти самого незанятого

Select Surname, MIN(COUNT(t.date\_turn))

From Employees AS e

JOIN Tasks AS t ON e.id\_empl = t.id\_empl

SELECT surname

FROM Employees

WHERE id\_employee = (

SELECT id\_empl

FROM Tasks

GROUP BY id\_empl

ORDER BY COUNT(\*)

LIMIT 1

);

Сколько дней выполняется каждый действующий проект

Select project\_name,

DATEDIFF(CURDATE(),

Select date\_delivery From Tasks

Where date\_turn IS NULL AND

)

From Projects;