

Разработка запросов к базе данных

План работы:

1. Стандартный язык запросов SQL.
2. Выборка данных.
3. Манипулирование данными.
4. Определение данных.
5. Примеры запросов к базе данных.
6. SQL в формах, отчетах и программах MS Access

Стандартный язык запросов SQL

1989г. ANSI (American National Standards Institute)

SQL – официальный международный стандарт не процедурного языка для формирования запросов к базам данных.

- Не обладает функциями полноценного языка разработки, а ориентирован на доступ к данным.
- Предоставляет развитые возможности как конечным пользователям, так и специалистам в области обработки данных.
- Многие современные СУБД могут подключаться к входным SQL-подсистемам с помощью технологии ODBC (Open Database Connectivity).
- Способен служить средством разработки масштабируемых систем типа «клиент-сервер».

Стандартный язык запросов SQL

Представление для пользователей:

- в явной синтаксической форме;
- В форме меню, диалоговых сценариев или заполняемых пользователем таблиц.

3

Основные функции:

- описание представления базы данных (ЯОД) - схема БД:
 - ✓ Описание структуры БД и налагаемых на неё ограничений целостности.
 - ✓ Ограничение доступа к данным и полномочий пользователям.
- выполнение операций манипулирования данными (ЯМД):
 - ✓ Добавление, изменение и удаление записей в таблицы.

Основные группы операторов языка SQL

1. Операторы определения данных (DDL)

CREATE – создание таблиц, индексов и представлений

ALTER – изменение описания таблиц, индексов и представлений

DROP – удаление таблиц, индексов и представлений

4

2. Операторы манипулирования данными (DML)

INSERT – добавление записей в таблицу

UPDATE – изменение данных в таблице

DELETE – удаление записей из таблицы

3. SELECT – оператор выборки данных

Выбирает данные не меняя содержимого БД

Дополнительные группы операторов

4. Средства администрирования

GRAND, REVOKE и т.д. – создание системы защиты данных с помощью паролей и разграничения доступа групп пользователей

5

5. Средства управления транзакциями

COMMIT, ROLLBACK, SAVEPOINT – завершение операций, сохранение промежуточного и возвращение к исходному состоянию.

Выборка данных

SELECT – отбор и сортировка данных из одной или нескольких связанных таблиц по заданному критерию, выполнение расчетов и преобразование выбранных данных с помощью специальных функций.

✓ Не меняет структуру и содержимое БД!

6

Синтаксис:

SELECT [**ALL** | **DISTINCT** | **TOP ЧИСЛО**] <Результирующий набор данных>

FROM <Описание таблиц и связей>

[**WHERE** <Условие отбора данных>]

[**GROUP BY** <Описание группировки>]

[**HAVING** <Условие для сгруппированных данных>]

[**ORDER BY** < Описание сортировки> [ASC|DESC]]

Выборка данных

SELECT [ALL | DISTINCT | TOP ЧИСЛО] <Результирующий набор данных>

ALL – отбор всех записей, удовлетворяющих условию отбора

DISTINCT – повторяющиеся строки не включаются в результат выполнения запроса

TOP ЧИСЛО – задаёт число выводимых строк (начиная с первой)

Выборка данных

`SELECT [ALL | DISTINCT | TOP ЧИСЛО] <Результирующий набор данных>`

– список полей или выражений, использующих агрегатные, математические и другие функции.

Элементы списка разделяются запятой.

Правила формирования списка:

8

- * для выбора всех столбцов
- **ИмяТаблицы.ИмяПоля**
- +, -, *, /, (), константы, стандартные функции
- агрегатные функции:
 - COUNT** – количество строк
 - SUM** – итоговые суммы
 - AVG** – среднее значение
 - MAX** – максимальное значение
 - MIN** – минимальное значение
- **As НовоеИмя**

Выборка данных

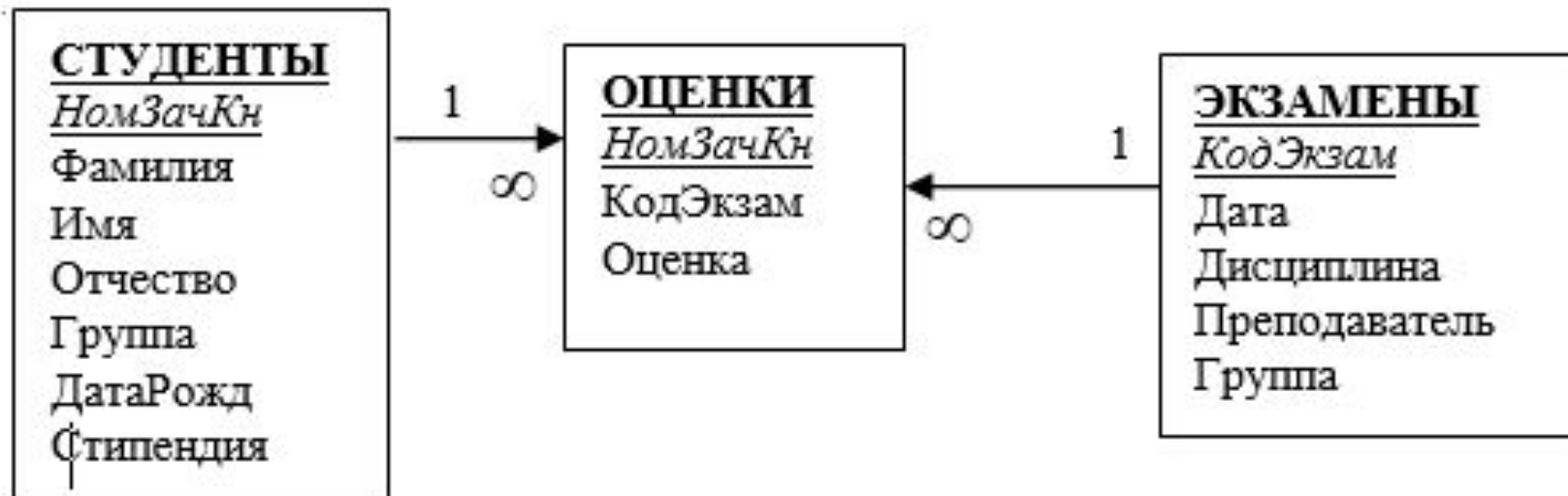
ПРИМЕР 1

Выбрать все данные из таблицы **ЭКЗАМЕНЫ**...

SELECT *

FROM ЭКЗАМЕНЫ

9



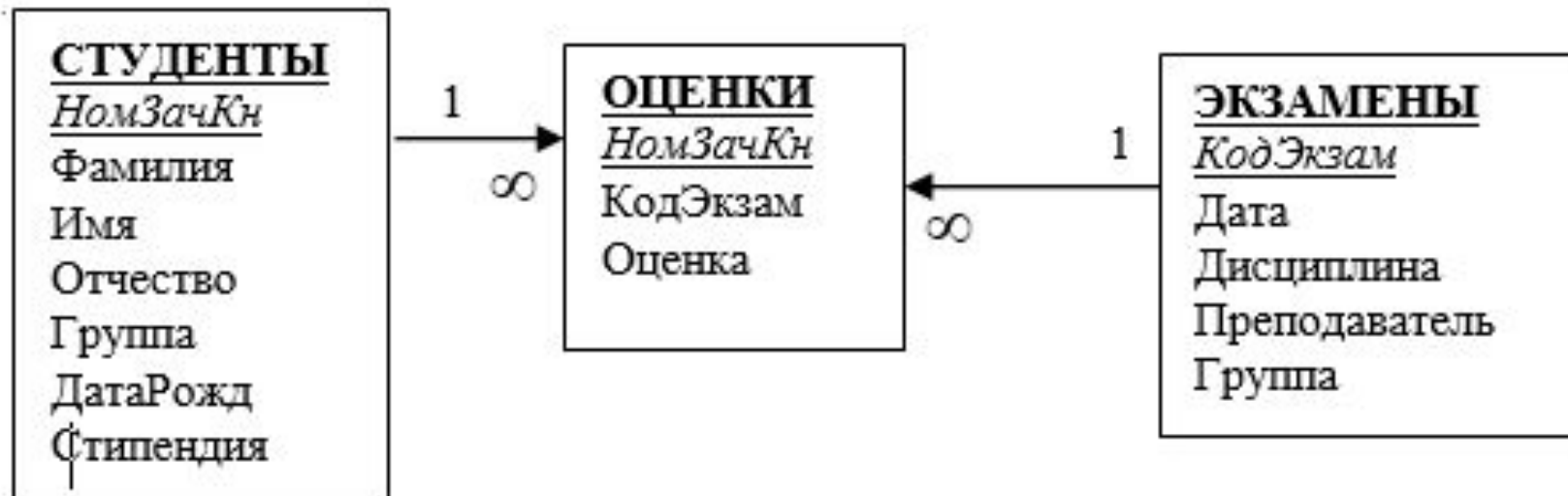
Выборка данных

ПРИМЕР 2

Выбрать данные из БД для получения документа “Расписание экзаменов” в порядке следования столбцов **Группа, Дисциплина, Дата...**

```
SELECT Группа, Дисциплина, Дата  
FROM ЭКЗАМЕНЫ
```

10



Выборка данных

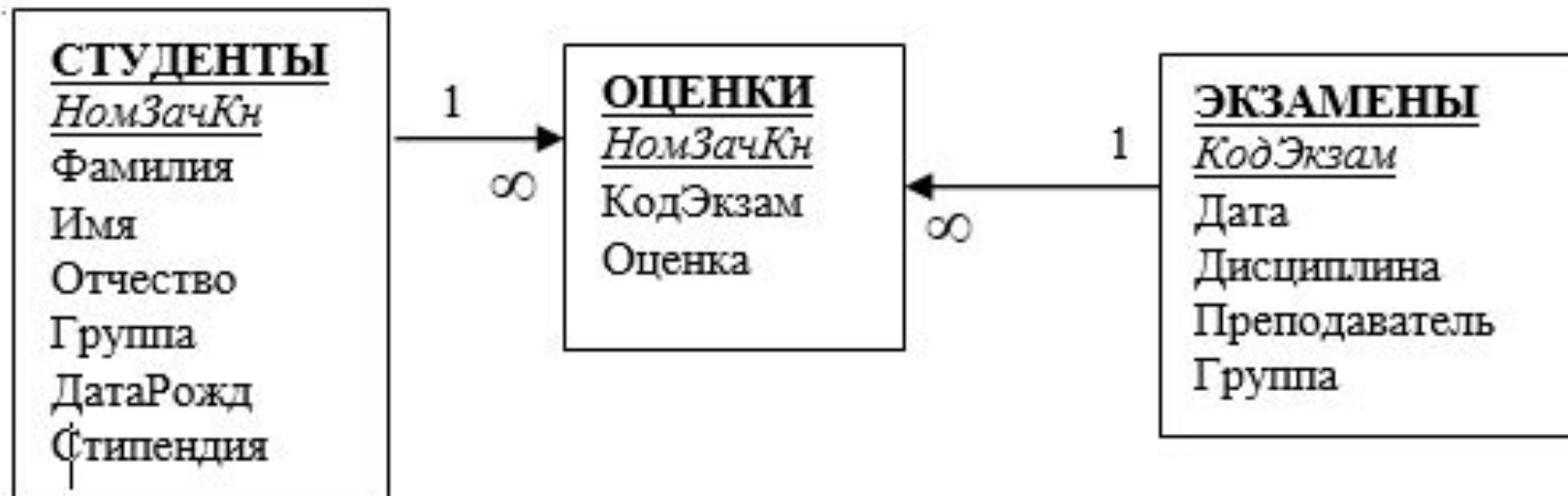
ПРИМЕР 3

Выбрать данные из БД для получения списка студентов в следующем виде в
Группа, Фамилия И.О., НомЗачКн, Стипендия....

```
SELECT Группа, Фамилия + ' ' + Left(Имя,1) + '.' + Left(Отчество,1) + '.'  
As 'Фамилия И.О.', НомЗачКн, Стипендия
```

```
FROM СТУДЕНТЫ
```

11



Выборка данных

SELECT [**ALL** | **DISTINCT** | **TOP ЧИСЛО**] <Результирующий набор данных>

FROM <Описание таблиц и связей>

[**WHERE** <Условие отбора данных>]

[**GROUP BY** <Описание группировки>]

[**HAVING** <Условие для сгруппированных данных>]

[**ORDER BY** < Описание сортировки> [ASC|DESC]]

– список полей или выражений, задающих условие сортировки.

ASC – сортировка в порядке возрастания (можно не указывать),

DESC – по убыванию.

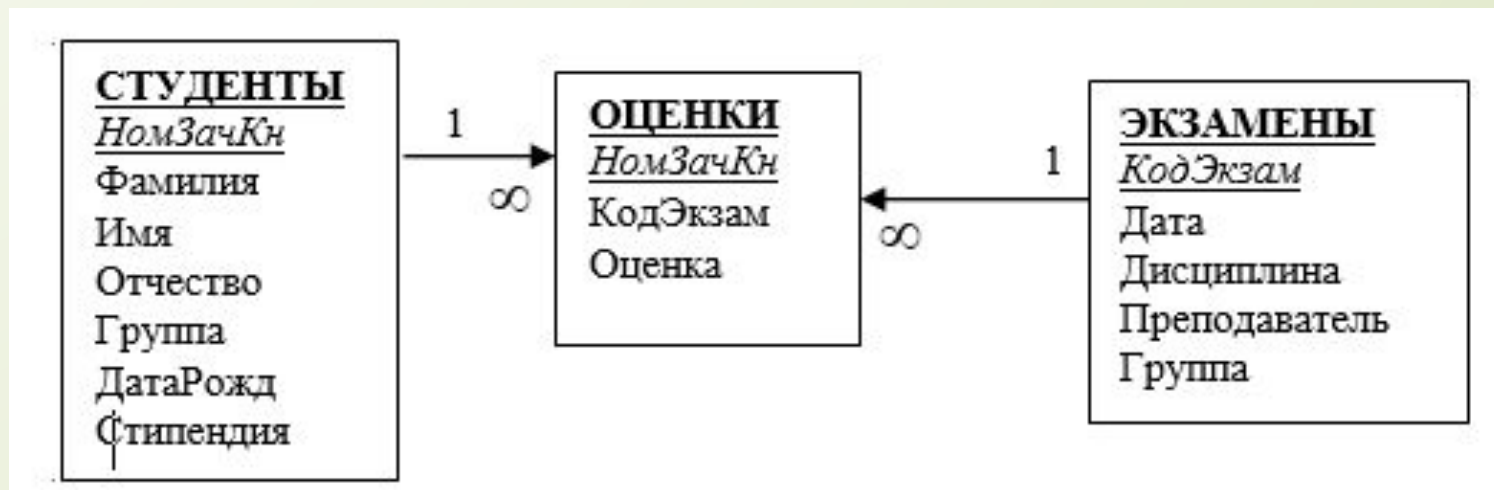
Выборка данных

ПРИМЕР 1

Выбрать все данные из таблицы **ЭКЗАМЕНЫ** в порядке следования их во времени, Сведения об экзаменах, которые спланированы в один день, расположить в порядке возрастания номеров групп.

```
SELECT *  
FROM ЭКЗАМЕНЫ  
ORDER BY Дата, Группа
```

13



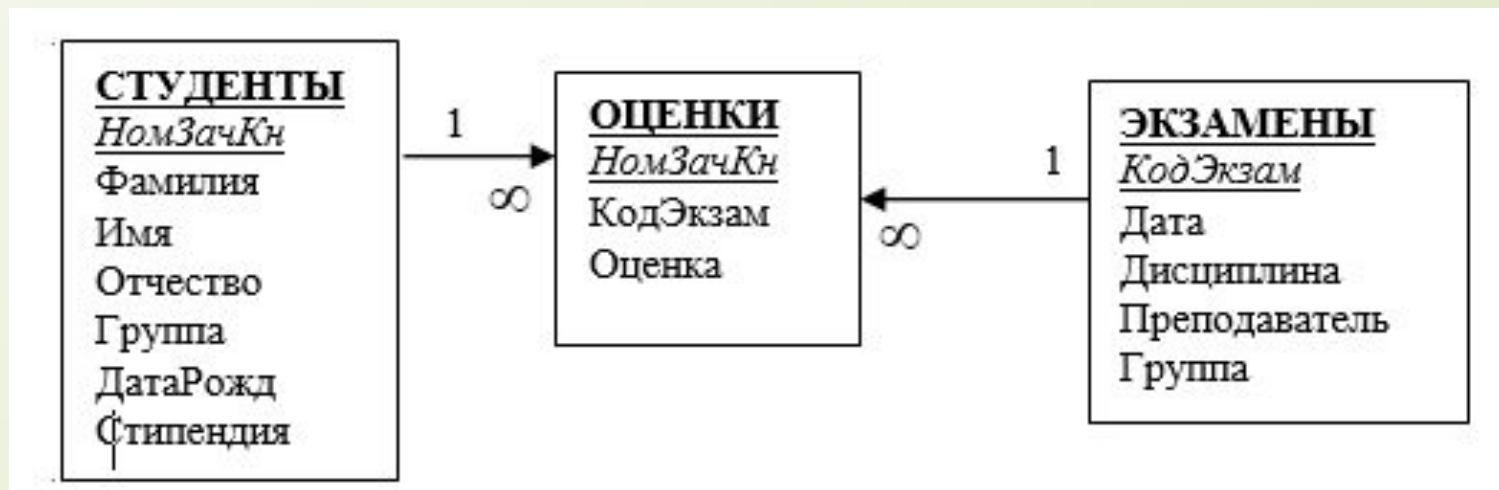
Выборка данных

ПРИМЕР 2

Выбрать данные из БД для получения документа “Расписание экзаменов” в порядке следования столбцов **Группа, Дисциплина, Дата**. Сортировку данных выполнить в порядке следования групп, а внутри одной группы – по дате.

14

```
SELECT Группа, Дисциплина, Дата  
FROM ЭКЗАМЕНЫ  
ORDER BY Группа, Дата
```



Выборка данных

ПРИМЕР 3

Выбрать данные из БД для получения списка студентов в следующем виде в **Группа, Фамилия И.О., НомЗачКн, Стипендия**. Сортировку выполнить в порядке возрастания номера группы, а внутри группы – в алфавитном порядке следования данных столбца **“Фамилия И.О.”**.

15

```
SELECT Группа, Фамилия + ' ' + Left(Имя,1) + ' ' + Left(Отчество,1) + ' ' AS  
'Фамилия И.О.', НомЗачКн, Стипендия
```

```
FROM СТУДЕНТЫ
```

```
ORDER BY Группа, Фамилия + ' ' + Left(Имя,1) + ' ' + Left(Отчество,1) + ' '
```

Выборка данных

SELECT [**ALL** | **DISTINCT** | **TOP ЧИСЛО**] <Результирующий набор данных>

FROM <Описание таблиц и связей>

[**WHERE** <Условие отбора данных>]

[**GROUP BY** <Описание группировки>]

[**HAVING** <Условие для сгруппированных данных>]

[**ORDER BY** < Описание сортировки>[ASC|DESC]]

- список таблиц (в этом случае связи задаются в части **WHERE**) или описание связей (внутренних или внешних) между таблицами.

Выборка данных

FROM <Описание таблиц и связей>

[WHERE <Условие отбора данных>]

<Условие отбора данных> - логическое выражение, описывающее критерий отбора записей из источников данных.

Правила формирования условных выражений:

17

- можно использовать знаки арифметических (+, -, *, /) и логических (=, <>, >, >=, <, <=, **AND**, **OR**, **NOT**) операций, круглые скобки, константы, поля и функции,

- к специальным операциям относятся: 1). проверка наличия значения в списке – **IN** (список значений). 2). проверка значения в заданном интервале – **BETWEEN N1 and N2**. 3). проверка на соответствие заданной маске – **LIKE** 'Маска', где символ % (* - для MS Access) заменяет любую последовательность символов, а символ подчёркивания (? - для MS Access) заменяет один любой символ,

- проверку на наличие в поле пустого, неопределённого значения можно выполнить с помощью инструкции – **IS NULL**, обратная операция – **IS NOT NULL** позволит определить те записи, где заданное поле заполнено

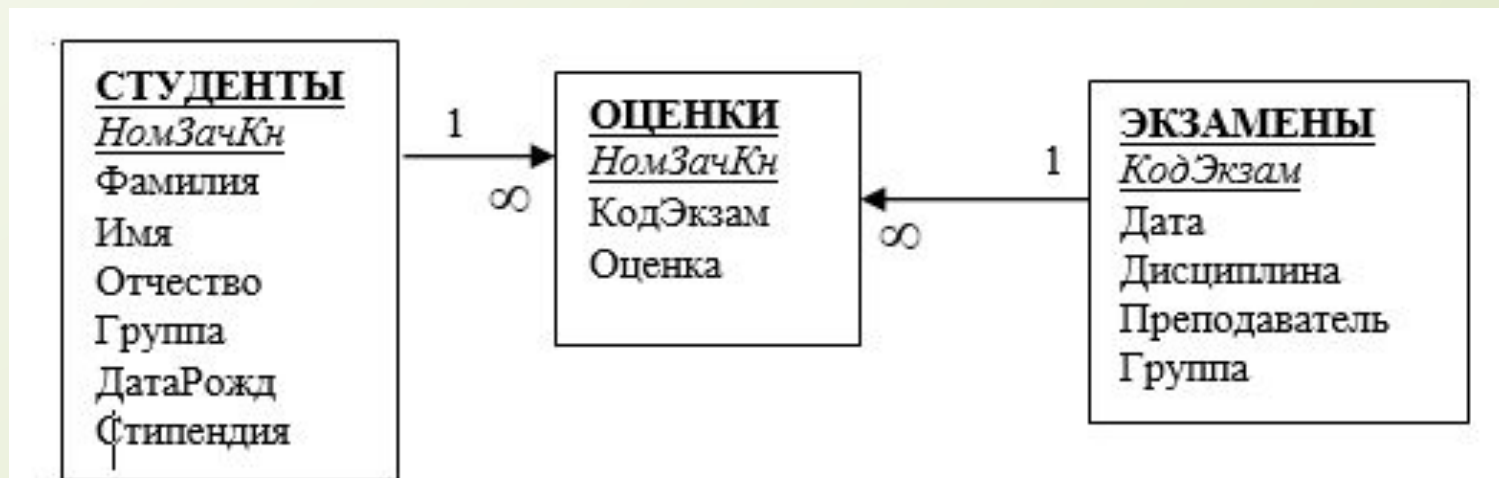
Выборка данных

ПРИМЕР 4

Получить в порядке возрастания номеров список групп, которые сдают экзамены в текущем месяце.

```
SELECT DISTINCT Группа  
FROM ЭКЗАМЕНЫ  
WHERE MONTH (Дата) = MONTH (DATE()) AND YEAR (Дата) =  
YEAR(DATE())  
ORDER BY Группа
```

18



Выборка данных

SELECT [**ALL** | **DISTINCT** | **TOP ЧИСЛО**] <Результирующий набор данных>

FROM <Описание таблиц и связей>

[**WHERE** <Условие отбора данных>]

[**GROUP BY** <Описание группировки>]

[**HAVING** <Условие для сгруппированных данных>]

[**ORDER BY** < Описание сортировки>[ASC|DESC]]

- список полей или выражений, задающих критерий формирования записей в группы (в группу включаются записи с совпадающими значениями столбцов, перечисленных в списке).
- логическое выражение, описывающее критерий отбора строк.

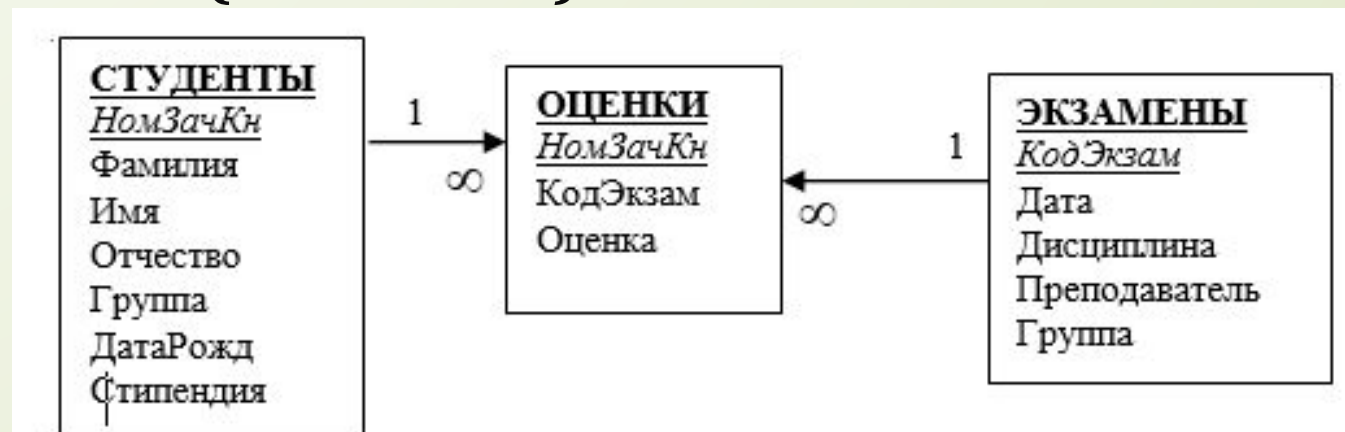
Выборка данных

ПРИМЕР 5

Подсчитать для каждой учебной группы количество студентов, получающих стипендию, а так же рассчитать сумму их стипендий, расположив строки результата в порядке убывания денежных сумм.

```
SELECT Группа , COUNT(*) AS Количество, SUM(Стипендия) AS Сумма
FROM СТУДЕНТЫ
WHERE Стипендия IS NOT NULL
GROUP BY Группа
ORDER BY SUM(Стипендия) DESC
```

20



Выборка данных

ПРИМЕР 6

Сведения о квартирах дома хранятся в таблице

ДОМ (Квартира, Подъезд, Этаж, Метров, Человек).

С помощью запроса рассчитать общую сумму оплаты услуг для каждой квартиры. Использовать следующие тарифы:

за отопление одного квадратного метра – 10 рублей,

за потребление воды одним человеком - 90 рублей,

за пользование лифтом взимается 70 рублей с каждой квартиры, расположенной на этаже выше 3-го.

Отсортировать строки по возрастанию номеров квартир.

Подъезд	Этаж	Квартира	Метров	Человек	Сумма
5	3	194	66	5	1 110 р.
5	4	195	49	3	830 р.

Выборка данных

ПРИМЕР 6

SELECT *Подъезд, Этаж, Квартира, Метров, Человек, Метров*10 +
Человек*90 + ИФ(Этаж>3, 70, 0) AS Сумма*

FROM *ДОМ*

ORDER BY *Квартира*

22

Подъезд	Этаж	Квартира	Метров	Человек	Сумма
5	3	194	66	5	1 110 р.
5	4	195	49	3	830 р.

Выборка данных из нескольких таблиц

FROM <Описание таблиц и связей>

[WHERE <Условие отбора данных>]

В среде СУБД Microsoft Access и Microsoft SQL Server внутренние и внешние связи между двумя таблицами можно задать с помощью следующей инструкции:

23

Таблица1 { INNER | LEFT | RIGHT } JOIN Таблица2

ON Таблица1.ПолеСвязи = Таблица2.ПолеСвязи

- **INNER** указывает на внутреннюю связь, при которой в результирующий набор выбираются только те записи, в которых значения полей связи совпадают.
- Внешнее соединение таблиц (**LEFT** – левое, **RIGHT** – правое) позволяет включить в результат запроса все строки из одной таблицы (**LEFT** – из Таблицы1, **RIGHT** – из Таблицы2) и соответствующие им строки из второй таблицы.

Выборка данных из нескольких таблиц

ПРИМЕР 7

Вывести список студентов (**Фамилия, Имя, Группа**), которые получили отличные оценки (сортировка по номеру группы, а внутри группы – в алфавитном порядке фамилий).

24

```
SELECT DISTINCT Фамилия, Имя, Группа  
FROM СТУДЕНТЫ INNER JOIN ОЦЕНКИ ON СТУДЕНТЫ.НомЗачКн =  
ОЦЕНКИ.НомЗачКн  
WHERE Оценка=5  
ORDER BY Группа, Фамилия, Имя
```


Выборка ДАННЫХ из нескольких таблиц

ПРИМЕР 8

Рассчитать средний балл сдачи экзаменов студентами 31 группы и представить данные в следующем виде (сортировку выполнить в алфавитном порядке фамилий).

```
SELECT Фамилия + ' ' + Имя As 'Фамилия Имя', ОЦЕНКИ.НомЗачКн,  
AVG(Оценка) As 'Средний балл'  
FROM СТУДЕНТЫ INNER JOIN ОЦЕНКИ ON СТУДЕНТЫ.НомЗачКн =  
ОЦЕНКИ.НомЗачКн  
WHERE Оценка=5  
ORDER BY Группа, Фамилия, Имя
```

25

Фамилия Имя	НомЗачКн	Средний балл
Иванов Павел	123	4,5

Выборка данных из нескольких таблиц

ПРИМЕР 9

Салон оказывает услуги своим клиентам по ценам действующего прейскуранта. Данные по учёту хранятся в двух таблицах: **ЦЕНЫ** (Услуга, Цена) и **РАБОТА** (Дата, Время, Мастер, Услуга).

С помощью запроса определите

26

- а). кто из мастеров сегодня выполнил услуг на большую сумму,
- б). какой вид услуг был самым популярным в прошлом году.

Выборка данных из нескольких таблиц

ПРИМЕР 9

а).

```
SELECT Мастер, SUM(Цена) As Сумма  
FROM ЦЕНЫ INNER JOIN РАБОТА ON ЦЕНЫ.Услуга = РАБОТА.Услуга  
WHERE Дата = DATE()  
GROUP BY Мастер  
ORDER BY SUM(Цена) DESC
```

27

б).

```
SELECT РАБОТА.Услуга, COUNT(*) As Число  
FROM ЦЕНЫ INNER JOIN РАБОТА ON ЦЕНЫ.Услуга = РАБОТА.Услуга  
WHERE YEAR(Дата) = YEAR(DATE())-1  
GROUP BY РАБОТА.Услуга  
ORDER BY COUNT(*) DESC
```

Манипулирование данными

INSERT

Добавление одной или нескольких записей с заполнением значениями всех или только некоторых полей таблицы.

а). добавление одной записи с заданными значениями в полях

28

INSERT INTO <Имя таблицы> [(Список полей)] **VALUES** (Список значений)

Пример.

Добавить новую запись в таблицу **ЭКЗАМЕНЫ**

INSERT INTO ЭКЗАМЕНЫ (КодЭкзам, Дата, Дисциплина, Группа)

VALUES (1245, #12.06.2006#, 'Базы данных', 35)

Манипулирование данными

INSERT

б). добавление одной или нескольких записей, отобранных из другой таблицы

INSERT INTO <Имя таблицы> [(Список полей)] <инструкция **SELECT**>

29

Пример.

Добавить в таблицу **АРХИВ** из таблицы **СТУДЕНТЫ** некоторые сведения о выпускниках факультета (т.е. о студентах с номером группы **> 50**).

INSERT INTO АРХИВ (НомЗачКн, Фамилия, Имя, Отчество, Группа)

SELECT НомЗачКн, Фамилия, Имя, Отчество, Группа

FROM СТУДЕНТЫ **WHERE** Группа>50

Манипулирование данными

UPDATE

Обновление значений полей во всех или нескольких записях, удовлетворяющих заданному условию.

30

UPDATE <Имя таблицы>

SET <Поле1> = <выражение1>, <Поле2> = <выражение2>, ...

[**WHERE** <Условие отбора данных>]

Манипулирование данными

UPDATE

Пример 1.

Увеличить все цены прейскуранта (таблица **ЦЕНЫ**) на 5%

```
UPDATE ЦЕНЫ SET Цена = Цена*1.05
```

Пример 2.

Заменить в поле Жанр таблицы **ФИЛЬМЫ** значение 'Триллер' на 'Ужасы'.

```
UPDATE ФИЛЬМЫ SET Жанр = 'Ужасы' WHERE Жанр = 'Триллер'
```

Манипулирование данными

DELETE

Удаление всех или нескольких записей, удовлетворяющих заданному условию.

32

DELETE <Имя таблицы>

[WHERE <Условие отбора данных>**]**

Манипулирование данными

DELETE

Пример 1.

Удалить все сведения о заказах, выполненных в прошлом году

DELETE FROM Заказы

WHERE YEAR(ДатаВыполн)=YEAR(DATE())-1

Пример 2.

Удалить все сведения о выпускниках из таблицы **СТУДЕНТЫ** (т. е. о студентах с номером группы **> 50**).

DELETE FROM СТУДЕНТЫ WHERE Группа>50