

## Лабораторная работа № 4.

### Выборка данных

#### **Цель работы**

Целью выполнения данной лабораторной работы является изучение оператора SELECT языка SQL; получение навыков его использования.

#### **Теоретические сведения**

Самый мощный оператор языка SQL — это оператор SELECT. Он служит для выборки данных, соответствующих указанным критериям и первичного форматирования результатов этой выборки. Приведем его полный синтаксис.

```
SELECT * | {[DISTINCT | ALL] {Поле данных} [, ...]}  
FROM {таблица [псевдоним]} [, ...]  
[WHERE условие]  
[GROUP BY {поле | Integer} [, ...]]  
[HAVING условие]  
[{UNION [ALL] | INTERSECT | EXCEPT SELECT-команда}] [...]  
[ORDER BY {поле | Integer [ASC | DESC]} [, ...]]  
[INTO {TEMP | SCRATCH} таблица];
```

SELECT возвращает как результат строки данных, которые создаются из данных, содержащихся в указанных исходных таблицах данных. С помощью SELECT можно, например, выбирать определенные строки из одной таблицы данных, или суммировать элементы этой таблицы, или же создавать новую таблицу-результат из многих исходных.

DISTINCT указывает базе данных исключать дубликаты из результата команды SELECT.

```
SELECT DISTINCT vnum  
FROM torder;
```

Здесь выбираются все различные (без дубликатов) элементы столбца

vnum таблицы torder.

С помощью FROM сообщают базе данных из каких таблиц выбирать данные.

```
SELECT vnum  
FROM torder;
```

Здесь выбираются все элементы поля vnum таблицы torder.

С помощью WHERE ограничивают выбор данных из указанных таблиц.

```
SELECT tkunden.knum, tverkauf.vnum, tverkauf.prov  
FROM tkunden, tverkauf  
WHERE tkunden.vnum=tverkauf.vnum;
```

Результатом является здесь таблица из строк, состоящих из поля knum таблицы tkunden и полей vnum и prov таблицы tverkauf, где с помощью WHERE указывается как находить строки таблицы tverkauf соответствующие строкам из таблицы tkunden.

```
GROUP BY {поле | Integer} [, ...]
```

GROUP BY используется вместе с функциями агрегирования данных для того, чтобы последние действовали на определенные группы строк результирующей таблицы данных. На месте имен полей могут быть использованы их порядковые номера в списке полей результирующей таблицы.

```
SELECT vnum, MAX(preis)  
FROM torder  
GROUP BY vnum;
```

Здесь выбираются в качестве результата все максимальные значения поля preis про найденную группу строк с идентичным значением поля vnum.

HAVING используется вместе с GROUP для того чтобы выбирать только определенные группы строк данных, которые удовлетворяют указанному условию.

```
SELECT vnum, MAX(preis)
```

```
FROM torder
GROUP BY vnum
HAVING AVG(preis) > 10
```

Здесь выбираются все максимальные значения поля `preis` про найденную группу строк данных с идентичным `vnum`, удовлетворяющую дополнительному условию, что средняя величина `preis` больше 10.

```
ORDER BY { поле | Integer [ ASC|DESC ] } [, ...]
```

`ORDER BY` сортирует строки результирующей таблицы данных. Если `ORDER BY` используется внутри `GROUP BY`, то строки сортируются внутри каждой группы результирующих строк. Вместо имен полей могут быть использованы их порядковые номера в списке полей результирующей таблицы. `ASC` сортирует данные в восходящем порядке, `DESC` — в обратном.

```
SELECT tkunden.knum, tverkauf.vnum, tverkauf.prov
FROM tkunden, tverkauf
WHERE tkunden.vnum=tverkauf.vnum
order by tkunden.knum asc;
```

`UNION` используется для того, чтобы объединять результаты двух или более команд `SELECT`. Поля всех результирующих таблиц должны быть совместимыми. Если `ALL` не использовано, то дубликаты исключаются из объединения результатов.

```
SELECT vnum, vname
FROM tverkauf
WHERE stadt='Berlin'
UNION
SELECT knum, kname
FROM tkunden
WHERE stadt='Berlin'
ORDER BY 1;
```

Здесь результатом является таблица, содержащая все `vnum` и `knum` таблицы `tverkauf` так же, как и все `vnum` и `kname` таблицы `tkunden`, где поле `stadt` имеет значение 'Berlin'. Результат сортируется по первому столбцу и дубликаты исключаются.

`INTERSECT` используется для того, чтобы объединять результаты двух или более команд `SELECT`, при этом в результат включаются только строки содержащиеся в первой команде `SELECT`, как и во второй. Поля всех результирующих таблиц должны быть совместимыми. Если `ALL` не использовано, то дубликаты исключаются из объединения результатов.

```
SELECT vnum, vname
FROM tverkauf
WHERE stadt='Berlin'
INTERSECT
SELECT knum, kname
FROM tkunden
WHERE stadt='Berlin'
ORDER BY 1;
```

Здесь интересуют все строки первой результирующей таблицы, которые так же содержатся и во второй. Дубликаты исключаются из результата.

`EXCEPT [ALL] SELECT-команда`

Важно: в некоторых диалектах SQL используется вместо `EXCEPT` `MINUS` или `DIFFERENCE`, синтаксис же от этого не меняется.

`EXCEPT` используется для того, чтобы объединять результаты двух или более команд `SELECT`, при этом в результат включаются только строки, содержащиеся в первой команде `SELECT` и не содержащиеся во второй. Поля всех результирующих таблиц должны быть совместимыми. Если `ALL` не использовано, то дубликаты исключаются из объединения результатов.

```
SELECT vnum, vname
```

```
FROM tverkauf
WHERE stadt='Berlin'
EXCEPT
SELECT knum, kname
FROM tkunden
WHERE stadt='Berlin'
ORDER BY 1;
```

Здесь интересуют все строки первой результирующей таблицы, которые не имеют дубликатов во второй.

INTO {TEMP | SCRATCH} таблица

INTO TEMP используется для создания вспомогательных таблиц в активной базе данных, которые автоматически удаляются из последней после окончания связи с ней. Часто используется для оптимизации команд SQL. INTO SCRATCH отличается от INTO TEMP тем, что временная таблица фрагментируется по всем дискам.

### ***Ход выполнения работы***

1. Изучить теоретические сведения.
2. Реализовать и протестировать запросы, возвращающие указанный преподавателем набор данных. При необходимости предусмотреть сортировку и группирование записей результата.
3. Составить отчет о выполнении лабораторной работы.
4. Подготовить ответы на контрольные вопросы.

### ***Контрольные вопросы***

1. Для чего используется оператор SELECT? Приведите пример.
2. Для чего используется ключевое слово WHERE? Приведите пример.
3. Как осуществляется сортировка результата выборки данных?
4. Для чего используется группировка данных? В чем ее отличие от сортировки?

5. Для чего используется ключевое слово `HAVING`?
6. Какими способами можно объединять результаты двух и более запросов?
7. Как можно исключить записи-дубликаты из результата выборки?