

## Лабораторная работа №2

### *Создание запросов к базе данных*

В СУБД Oracle для создания запросов используется язык SQL. Создавать запросы можно в программе SQL\*Plus. Далее рассмотрим основные виды запросов.

#### *Запросы на выборку*

Для выбора информации из базы данных используется оператор выбора **select**.

Синтаксис данного оператора имеет вид:

SELECT [DISTINCT] <имя\_столбца> [AS <новое\_имя\_столбца>],...

FROM <имя\_схемы.имя\_таблицы>,...

[WHERE <условие\_выборки>]

[GROUP BY <имя\_столбца>,...]

[HAVING <условие\_выборки>].

- **DISTINCT** — исключает дубликаты записей и одновременно упорядочивает полученный в результате перечень;
- **SELECT** — определяет столбцы, которые необходимо вывести в качестве результата выполнения запроса. В предложении select могут содержаться операторы работы с числами, такие как /, \*, -, + и функции агрегирования:

**COUNT** — подсчитывает количество всех значений в столбце за исключением значения **NULL** и с учетом указания фраз **ALL** или **DISTINCT**;

**COUNT (\*)** — подсчет количества всех значений столбцов в группе;

**AVG** — определение среднего значения;

**SUM** - подсчет суммы всех значений группы. Если при этом получаемое значение выходит за пределы суммируемого типа данных, то выдается ошибка выполнения SQL-оператора;

**MAX** - определение максимального значения из группы; **MIN** - определение минимального значения из группы;

- **AS** — позволяет задать новое имя столбцу;
- **FROM** - определяет таблицы, используемые для выборки;
- **WHERE** - определяет условия отбора.

Для задания условия, кроме операторов =, O, >, <, >=, <=, AND, OR, NOT также могут использоваться следующие операторы:

**BETWEEN**, например: <имя\_столбца> BETWEEN x AND y, возвращает TRUE для значений от x до y;

**IN**, например: <имя\_столбца> IN (x,y,z), возвращает TRUE для значений x,y,z;

**LIKE** - возвращает TRUE для значений, совпадающих с указанной подстрокой символов. Например: <имя\_столбца> LIKE 'zzx';

**IS NULL**, например <имя\_столбца> IS NULL, возвращает TRUE, если значение равно NULL;

- **GROUP BY** - разбивает записи на группы и обобщает

значения для каждой из групп. Если в запросе указана фраза GROUP BY, то все имена столбцов, указываемые в списке для определения создаваемого результирующего набора, должны быть указаны с функциями агрегирования;

- **HAVING** — определяет условие отбора аналогично фразе WHERE, но применяется к строкам при использовании функций агрегирования.

Рассмотрим использование вышеприведенных операторов **на примерах:**

```
1. SQL> SELECT "NOM_STUDENTA", "GROUP"  
AS "NOM_GROUP"  
FROM " SYSTEM "."STUDENT_DATE"  
WHERE "GROUP" = 'CB-101';
```

Этот оператор выводит все номера студентов из группы CB-101, формируя новое имя столбца NOM\_GROUP для с

- ```
2. SQL> SELECT COUNT("SURNAME"), "GROUP" FROM " SYSTEM  
"."STUDENT_DATE" GROUP BY "GROUP" HAVING COUNT("SURNAME")>20;
```
- Данный оператор подсчитывает количество студентов по группам и выводит результат только для тех групп в которых учится более 20 студентов.

### ***Запросы на модификацию данных***

Запрос на **добавление** записей в таблицу использует оператор INSERT. Формат оператора:

```
INSERT INTO <имя_схемы.имя_таблицы> [(<имя_столбца>, ...)] VALUES  
(значение, ...) [SELECT [DISTINCT] <имя_столбца>,... FROM  
<имя_схемы.имя_таблицы> [WHERE <условие_выборки>] [GROUP BY  
<имя_столбца>,...] [HAVING <условие_выборки>]].
```

### **Пример.**

```
SQL> INSERT INTO " SYSTEM "."STUDENT_DATE" VALUES (1114, Глушков',  
'ул.Менделеева 57-43', TO_DATE('06/12/1992','MM/DD/YYYY'), 'CB-101'); В
```

приведенном выше примере список столбцов таблицы опущен, т.к. вставка новых значений подразумевается во все столбцы, причем вставка данных в столбцы должна происходить в том порядке, который был установлен при создании таблицы. Здесь также рассмотрен способ вставки значения даты в таблицу с помощью функции TO\_DATE, т.к. иногда возможна проблема определения формата даты с которым работает определенная СУБД. В следующем примере указан порядок следования только тех столбцов, в которые происходит вставка данных.

### **Пример.**

```
SQL> INSERT INTO "SYSTEM"."STUDENT_DATE"  
("SURNAME", "GROUP", "NOM_STUDENTA")  
VALUES (Глушков', 'CB-101', 1114);
```

Эти два примера вставляют единственную строку в указанную таблицу. Рассмотрим случай, когда нужно вставить сразу несколько записей.

### **Пример.**

```
SQL> INSERT INTO "SYSTEM"."KURS_RABOTA"  
("NOM_STUDENTA")  
SELECT "NOM_STUDENTA"  
FROM " SYSTEM"."STUDENT_DATE";
```

Этот оператор копирует значения столбца "NOM\_STUDEN-ТА" из таблицы "SYSTEM"."STUDENT\_DATE" в таблицу "SYSTEM". "KURS\_RABOTA".

Запрос на **обновление** использует оператор UPDATE и применяется для изменения значений в одной или нескольких записях указанной таблицы. Формат оператора:

```
UPDATE <имя_схемы.имя_таблицы>  
SET <имя_столбца>=<выражение>, ...  
[WHERE <условие_выборки>].
```

**Пример.**

```
SQL> UPDATE "SYSTEM"."STUDENT_DATE"  
SET "SURNAME" = 'Глушкова' WHERE "SURNAME" = 'Иванова';
```

Этот оператор устанавливает изменение фамилии Иванова на Глушкову.

Запрос на **удаление** использует оператор DELETE и предназначен для удаления одной или нескольких записей из таблицы. Формат оператора:

```
DELETE FROM <имя_схемы.имя_таблицы> [WHERE <условие_выборки>].
```

Пример.

```
SQL> DELETE FROM "SYSTEM"."STUDENT_DATE"  
WHERE "SURNAME" = Глушков';
```

Эта команда удаляет из таблицы "SYSTEM"."STUDENT\_DATE" все данные о студентах с фамилией Глушков.

***Выборка данных из нескольких таблиц, используя операции соединения***

Такие запросы обычно строятся на том, что у одной таблицы имеется первичный ключ, а у другой внешний ключ. Сравнивая значение первичного ключа с внешним ключом, получают объединение данных из таблиц. Для чтения данных из нескольких таблиц необходимо использовать операцию соединения - JOIN. Существует несколько видов соединения, которые используют во фразе FROM:

- CROSS JOIN - перекрестное соединение. Результат получается путем перекрестного перемножения двух таблиц.

**Пример.**

Для рассмотрения всех примеров данного раздела воспользуемся таблицами KURS\_RABOTA, STUDENT\_DATE с данными:

| NOM_STUDENTA | NAME_KURS_RAB            |
|--------------|--------------------------|
| 1111         | Система наполнения сайта |
| 1112         | Атака на SPAM - сервер   |
| 1113         | Система мониторинга сети |

| NOM_STUDENTA | SURNAME         | ADRESS           | DATE        | GROUP  |
|--------------|-----------------|------------------|-------------|--------|
| 1111         | Иванов К.М.     | Мира 11-45       | 13-сен-1986 | св-304 |
| 1112         | Петров К. О.    | Бородина 67/3-21 | 11-янв-1986 | св-102 |
| 1113         | Семенов Л.Д.    | Ленина 35-67     | 06-авг-1983 | сб-303 |
| 1114         | Колмагоров Т.Г. | Бородина 23/2-98 | 01-янв-1985 | св-304 |
| 1115         | Сенков Т.А.     | Вавилова 123-78  | 27-ноя-1984 | сб-104 |

SQL> SELECT "SURNAME", "NAME\_KURS\_RAB" FROM "SYSTEM"."STUDENT\_DATE" CROSS JOIN "SYSTEM"."KURS\_RABOTA";

Результат этого запроса будет таким:

| SURNAME         | NAME_KURS_RAB            |
|-----------------|--------------------------|
| Иванов К.М.     | Система наполнения сайта |
| Петров К.О.     | Система наполнения сайта |
| Семенов Л.Д.    | Система наполнения сайта |
| Колмогоров Т.Г. | Система наполнения сайта |
| Сенков Т.А.     | Система наполнения сайта |
| Иванов К.М.     | Атака на SPAM -- сервер  |
| Петров К.О.     | Атака на SPAM -- сервер  |
| Семенов Л.Д.    | Атака на SPAM -- сервер  |
| Колмогоров Т.Г. | Атака на SPAM -- сервер  |
| Сенков Т.А.     | Атака на SPAM -- сервер  |
| Иванов К.М.     | Система мониторинга сети |
| Петров К.О.     | Система мониторинга сети |
| Семенов Л.Д.    | Система мониторинга сети |
| Колмогоров Т.Г. | Система мониторинга сети |
| Сенков Т.А.     | Система мониторинга сети |

15 строк выбрано.

• NATURAL INNER JOIN - внутреннее соединение, используется по умолчанию. Соединяются только те строки из таблиц значения, которых совпадают.

### Пример.

SQL> SELECT "NOM\_STUDENTA", "SURNAME",  
"NAME\_KURS\_RAB"  
FROM "SYSTEM"."STUDENT\_DATE"  
NATURAL INNER JOIN "SYSTEM"."KURS\_RABOTA";

Результат этого соединения будет следующим:

| NOM_STUDENTA | SURNAME      | NAME_KURS_RAB            |
|--------------|--------------|--------------------------|
| 1111         | Иванов К.М.  | Система наполнения сайта |
| 1112         | Петров К.О.  | Атака на SPAM -- сервер  |
| 1113         | Семенов Л.Д. | Система мониторинга сети |

- NATURAL LEFT JOIN - левое внешнее соединение. При левом соединении создаются все строки из левой таблицы (т.е. таблицы стоящей слева от ключевого слова JOIN) и соответствующие им строки из правой.

### Пример.

```
SQL> SELECT "NOM_STUDENTA", "SURNAME",
"NAME_KURS_RAB"
FROM "SYSTEM"."STUDENT_DATE"
NATURAL LEFT JOIN "SYSTEM"."KURS_RABOTA";
```

Результат имеет следующий вид:

| NOM_STUDENTA | SURNAME         | NAME_KURS_RAB            |
|--------------|-----------------|--------------------------|
| 1111         | Иванов К.М.     | Система наполнения сайта |
| 1112         | Петров К.О.     | Атака на SPAM -- сервер  |
| 1113         | Семенов Л.Д.    | Система мониторинга сети |
| 1114         | Колмогоров Т.Г. |                          |
| 1115         | Сенков Т.А.     |                          |

- NATURAL RIGHT JOIN - правое внешнее соединение.

При правом соединении создаются все строки из правой таблицы (т.е. таблицы стоящей справа от ключевого слова JOIN) и соответствующие им строки из левой.

### Пример.

```
SQL> SELECT "NOM_STUDENTA", "SURNAME",
"NAME_KURS_RAB"
FROM "SYSTEM"."STUDENT_DATE"
NATURAL RIGHT JOIN "SYSTEM"."KURS_RABOTA";
```

Результатом соединения будет следующая таблица:

| NOM_STUDENTA | SURNAME      | NAME_KURS_RAB            |
|--------------|--------------|--------------------------|
| 1111         | Иванов К.М.  | Система наполнения сайта |
| 1112         | Петров К.О.  | Атака на SPAM -- сервер  |
| 1113         | Семенов Л.Д. | Система мониторинга сети |

- NATURAL FULL JOIN - полное внешнее соединение.

При полном внешнем соединении результирующая таблица будет содержать все строки, как из правой, так и из левой таблицы. Если существуют пустые значения, то им присваивается значение NULL.

### Пример.

```
SQL> SELECT "NOM_STUDENTA", "SURNAME",
"NAME_KURS_RAB"
FROM "SYSTEM"."STUDENT_DATE"
NATURAL FULL JOIN "SYSTEM"."KURS_RABOTA";
```

Результатом будет таблица:

| NOM_STUDENTA | SURNAME         | NAME_KURS_RAB            |
|--------------|-----------------|--------------------------|
| 1111         | Иванов К.М.     | Система наполнения сайта |
| 1112         | Петров К.О.     | Атака на SPAM -- сервер  |
| 1113         | Семенов Л.Д.    | Система мониторинга сети |
| 1114         | Колмогоров Т.Г. |                          |
| 1115         | Сенков Т.А.     |                          |

- UNION — соединение объединения. При объединении запросов каждый из них должен содержать одинаковое количество столбцов одного и того же типа. Столбцам, по которым происходит объединение, присваиваются имена из первого запроса.

### **Пример.**

```
SQL> SELECT "SURNAME"  
FROM "SYSTEM"."STUDENT_DATE" UNION SELECT "NAME_KURS_RAB"  
FROM "SYSTEM"."KURS_RABOTA";
```

Результат этого объединения выглядеть будет следующим образом:

```
SURNAME  
-----  
Иванов К.М.  
Атака на SPAM -- сервер  
Колмогоров Т.Г.  
Петров К.О.  
Семенов Л.Д.  
Сенков Т.А.  
Система мониторинга сети  
Система наполнения сайта  
  
8 строк выбрано.
```

### ***Вложенные запросы***

Для выполнения сложных запросов можно использовать оператор select внутри предложения where другого оператора select, **например:**

```
SQL> SELECT "SURNAME", "GROUP"  
FROM "SYSTEM"."STUDENT_DATE"  
WHERE  
"ADRESS" LIKE 'Ленина%' AND "NOM_STUDENTA" IN  
(SELECT "NOM_STUDENTA",  
FROM "SYSTEM"."KURS_RABOTA"  
WHERE "NAME_KURS_RAB" LIKE 'Система%');
```

В данном примере мы объединили два оператора SQL, используя вложенный запрос. В результате выполнения данного оператора сначала произойдет выборка данных из таблицы KURS\_RABOTA, удовлетворяющих условию, что названия курсовых должны начинаться со слова "Система", затем из полученных данных выберутся из таблицы STUDENT\_DATE только те, которые удовлетворяют условию - проживание на улице Ленина.

## **Задание к лабораторной работе №2**

Создайте SQL-операторы, выполняющие указанные ниже действия к базам данных, разработанным в лабораторной работе №1. В зависимости от реализации базы данных в предыдущей работе добавьте два собственных запроса.

### **Вариант №1**

1. Подсчитать количество товаров имеющих цену свыше 1500 руб.
2. Для постоянных клиентов старше 50 лет сделать скидку на заказ 12%.
3. Удалите из таблицы все записи о заказах сделанных до 31.12.99
4. Вывести данные о покупателях не живущих в России.

5. Создайте запись о новом поставщике.
6. Определите максимальную стоимость товара за 2003-2004 год.

### **Вариант №2**

1. Определить суммарную величину излишков и недостачи по складу продовольственных товаров.
2. Найти по наименованию товара печенье "Юбилейное" адрес предприятия поставившего этот товар
3. Удалите запись о товаре кофе "Вдохновение".
4. Создайте запрос, в котором содержались бы сведения о фирмах поднявших цену на товар более чем на 50 руб. за последний год.
5. Определить список товаров срок годности, которых истекает в декабре 2005 года.
6. Создайте запись о новом магазине.

### **Вариант №3**

1. Найти директора предприятия по фамилии Иванов и увеличить его кредит на 100000 руб.
2. Определить разнообразие товара, выпускаемого омскими предприятиями.
3. Для предприятия "Стройинвест" определить количество купленного товара.
4. Определить предприятия Омска, покупающие товар с кодом 9874...778.
5. Получить сведения о товарах купленных предприятиями в апреле 2004 года.
6. Удалить данные о предприятии "Комфорт" и товарах, которые оно выпускало.

### **Вариант №4**

1. Найти предприятия, заказавшие "краска белила цинковые".
2. Определить общую сумму по следующим полям "оборот за месяц", "остаток на конец месяца", "остаток на начало месяца".
3. Вывести те товары, у которых "остаток на конец месяца" меньше 10000.
4. Определить предприятия, заказавшие на складе товары, у которых "остаток на начало месяца" равен 56... 789.
5. Получить сведения о предприятиях, заказавших товар по минимальной цене и не заказавших товар по другой цене.
6. Перенести все товары, хранившиеся на складе №1 на склад №4.

### **Вариант №5**

1. Вывести список сотрудников кафедры "Экономика и управление", рассчитав для них заработную плату с учетом оклада и установленных надбавок.
2. Вывести список сотрудников кафедры "Информационные технологии", чей день рождения приходится на октябрь.

3. Получить список сотрудников, имеющих ученое звание "доцент".
4. Получить сведения о сотрудниках, имеющих окончания в фамилиях "ова" или "ов".
5. Вывести данные, в которых содержатся данные о сотрудниках, имеющих больше трех детей.
6. Создать запись о новом работнике с почасовой оплатой, работающим на кафедре кибернетики.

#### **Вариант №6**

1. Найти книги, у которых год издания 197...
2. Создать запрос, в результате которого выводится книга, имеющая наибольшую цену.
3. Получить список книг, находящихся в пользовании более 10 лет и чаще всего востребованных читателями.
4. Вывести все данные о книгах серии "фантастика".
5. Вывести все фамилии и телефоны читателей, которые просрочили сдачу книг более, чем на 10 дней.
6. Удалить данные о читателях записанных в библиотеку, но не посещавших библиотеку более двух лет.

#### **Вариант №7**

1. Перечислить лекарства, купленные за последние два дня.
2. Удалить тех заказчиков, заказы которых были выполнены.
3. Создать запрос, который выводит среднюю стоимость всех лекарств.
4. Создать запрос, который содержит фамилию заказчиков, заказавших самое дорогое лекарство, и если он пенсионер сделать для них скидку 2%.
5. Выполнить запрос, в котором по показанию к применению выводится список лекарств.
6. Оформить заказ покупателя на лекарство "Инсулин".

#### **Вариант №8**

1. Подсчитать количество заказчиков, заказавших фотографию размером 10x15.
2. Создать запрос, который позволял бы получить сведения о заказчиках и фотографах, которые будут выполнять данный заказ.
3. Вывести данные о заказчиках, дата выполнения заказа у которых 21.06.05 или 22.06.05.
4. Создать запрос, в котором указывается ФИО заказчика и цена заказа (цена расходных материалов + 10% за работу).
1. Введите данные о новом заказчике.
6. Для клиентов сделавших заказ в праздничные дни сделать скидку на 2%.

#### **Вариант №9**

1. Вывести наименование тех растений, которые были удобрены месяц назад.
2. Определить код растений, нуждающихся в интенсивной поливке (2 раза в день).
3. Составить запрос, в котором дата полива изменена на



один день позже.

4. Создать запрос, в котором выводится список тех растений, которые требуют максимального освещения.
5. Увеличить на 50% количество растений для посадки в этом году, которые цветут в начале марта.
6. Удалить данные о растениях, выращивание которых не приносит доход (минимум продаж; за прошедший год).

### **Вариант №10**

1. Вставить записи о пассажирах, купивших билеты на сегодня и завтра.
2. Найти вагон, в котором максимальное количество свободных мест.
3. Создать запрос, который позволял бы определить время, проведенное пассажиром в пути, если он едет из начального пункта следования до конечного.
4. Составить запрос, в котором была бы определена сумма выручки от проданных билетов за прошедший месяц.
5. Получить записи, в которых содержится список поездов, имеющих наибольший спрос у пассажиров.
6. Повысить стоимость билетов на весь железнодорожный транспорт с 1 января 2006 года на 15%.

### **Вариант №11**

1. Найти фамилии абонентов, у которых номер телефона заканчивается на цифру 7.
2. Для клиента, у которого количество звонков в месяц наибольшее, уменьшить стоимость услуг на 10%.
3. Создать запрос, который содержит: код абонента, стоимость предоставляемой услуги и общую среднюю стоимость предоставляемой услуги за единицу.
4. Создать запрос, в результате которого выводится список абонентов, занимающих должность "менеджер" и работающих в данной службе МТС.
5. На все входящие звонки для любого абонента сделать скидку в апреле месяце.
6. Вывести платежную ведомость для Павлова за текущий месяц.

### **Вариант №12**

1. Выбрать ФИО, адрес клиента, не бравших кредит.
2. Вывести список клиентов, у которых фамилия начинается на С и относящихся к филиалу, расположенному по улице Нефтезаводская.
3. Создать запрос, в результате которого выводится № счета клиента, имеющего кредит 500000 руб.
4. Вывести данные о клиентах, имеющих льготы по кредиту и уменьшить этот кредит на 1%.
5. Удалить данные о клиенте, у которого на счете 0 руб.
6. Получить данные о клиенте, который имеет максимальный счет.

### **Вариант №13**

1. Определить наименование товаров продающихся в отделе "Консервы".
2. Создать запрос, в результате которого выводится список

просроченных товаров.

1. Найти количество товаров молочного ассортимента.
2. Создать запрос о товарах по цене меньше 30 руб.
3. Создать запрос, в котором содержится количество покупателей, посетивших магазин за последние 2 дня (каждый покупатель покупает в среднем товар в количестве 5 штук).
4. Определить список товаров, которых осталось на прилавках меньше 10% от товара с максимальным количеством.