Базы данных

Язык запросов SQL.
Команда SELECT (продолжение)

Использование фразы HAVING

Если необходимо вывести не все записи, полученные в результате группировки (GROUP BY), то условие на группы можно указать во фразе HAVING (но не во фразе WHERE).

Пример. Список отделов, в которых работает больше пяти человек:

```
select  depno,  count(*), 'человек(а)'
        from  emp
        group by  depno
        having  count(*)>5;
```

Правило: нельзя указывать агрегирующие функции в части WHERE – это синтаксическая ошибка!

Задание: вывести список отделов, в которых средняя зарплата больше 30000 рублей.

```
select  depno,  avg(salary)
        from  emp
        group by  depno
        having  avg(salary) > 30000;
```

Операции реляционной алгебры

Унарные операции:

- **селекция** – выбор из таблицы подмножества строк по условию.
  Например, список сотрудников 5-го отдела:

```
select *
    from emp
    where depno = 5;
```

- **проекция** – выбор из таблицы подмножества столбцов.
  Например, сведения о должности и зарплате сотрудников:

```
select  distinct name,  post,  salary
    from  emp;
```

Бинарные операции реляционной алгебры

Бинарные операции РА:

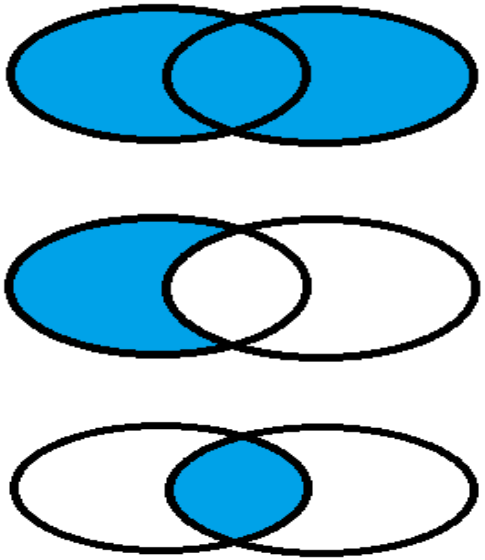
- **разносхемные** – применяются к любым двум отношениям.
- **односхемные** – применяются к односхемным отношениям. Исходные отношения должны иметь одинаковое количество столбцов одинаковых (или сравнимых) типов. ***Сравнимыми*** считаются типы, относящиеся к одному и тому же семейству данных (в таблице полужирным шрифтом выделены базовые типы).

Семейства типов данных Oracle:

Числовые:	**Символьные:**	**Календарные:**
DEC, DECIMAL, DOUBLE PRECISION, FLOAT, INT, INTEGER, **NUMBER,** NUMERIC, REAL, SMALLINT	CHAR, CHARACTER, LONG, LONG RAW RAW, ROWID, STRING, VARCHAR, **VARCHAR2**	**DATE**

Бинарные односхемные операции РА

- ✓ **Объединение** двух односхемных отношений содержит все строки исходных отношений без повторов.
- ✓ **Разность** двух односхемных отношений содержит все строки первого отношения, не входящие во второе отношение (без повторов).
- ✓ **Пересечение** двух односхемных отношений содержит все строки, входящие и в первое, и во второе отношения (без повторов).



Добавим в нашу БД проектной организации таблицу "Архив должностей":

```
create table  archive (
        tabno    number(6) REFERENCES  emp,        -- ссылка на сотрудника
        name     varchar2(100)  not null,          -- ФИО сотрудника
        dbegin   date  not null,                   -- начало работы в должности
        post     varchar(50)  not null             -- должность
);
```

Операция объединения

Объединение реализуется с помощью специального ключевого слова **UNION** (или **UNION ALL**, если не нужно удалять повторы).

Примеры:

- Список сотрудников с телефонами или адресами (если нет телефона):

```
select depno, name, PHONE
    from emp where phone is not null
UNION ALL
select depno, name, ADR
    from emp where phone is null;
```

- Список сотрудников со всеми переводами с одной должности на другую:

```
select tabno, name, edate, post
    from emp
UNION ALL
select tabno, name, dbegin, post
    from archive
   order by 1, 3;
```

Разность отношений

Разность в Oracle реализуется с помощью специального ключевого слова **MINUS**.

Примеры:

- Список сотрудников 5-го и 8-го отделов, которые не являются инженерами:

```
select * from emp
    where depno IN (5, 8)
MINUS
select * from emp
    where post LIKE '%инженер%'
order by depno;
```

- Список сотрудников, которые не переводились на другие должности:

```
select tabno, name
    from emp
MINUS
select tabno, name
    from archive;
```

Пересечение отношений

Пересечение в Oracle реализуется с помощью специального ключевого слова **INTERSECT**.

Примеры:

- Список сотрудников 5-го и 8-го отделов, которые являются инженерами:

```
select * from emp
    where depno IN (5, 8)
INTERSECT
select * from emp
    where post LIKE '%инженер%'
order by depno;
```

- Список сотрудников, которые переводились на другие должности:

```
select tabno, name
    from emp
INTERSECT
select tabno, name
    from archive;
```

Применение односхемных операций РА

Задание 1: вывести список должностей, которые занимают (или занимали) сотрудники.

```
select post from emp
UNION
select post from archive;
```

Задание 2: вывести список должностей, на которые переназначены другие сотрудники.

```
select post from emp
INTERSECT
select post from archive;
```

Задание 3: вывести список должностей, которые в настоящее время не занимает ни один сотрудник.

```
select post from archive
MINUS
select post from emp;
```

Разносхемные операции РА

Декартово произведение (ДП): операция над двумя произвольными (возможно, разносхемными) отношениями. Результат ДП – все комбинации строк исходных отношений. Пример:

"Студенты"

Группа	*ФИО*
СТ-4/09	Рогов В.П.
СТ-4/09	Белова О.Г.

"Предметы"

Предмет
Базы данных
Сетевые технологии

"Оценки"

Оценка
удовл.
хор.
отл.

Декартово произведение: "Оценки студентов"

Группа	*ФИО*	*Предмет*	*Оценка*
СТ-4/09	Рогов В.П.	Базы данных	удовл.
СТ-4/09	Рогов В.П.	Базы данных	хор.
СТ-4/09	Рогов В.П.	Базы данных	отл.
СТ-4/09	Рогов В.П.	Сетевые технологии	удовл.
СТ-4/09	Рогов В.П.	Сетевые технологии	хор.
СТ-4/09	Рогов В.П.	Сетевые технологии	отл.
СТ-4/09	Белова О.Г.	Базы данных	удовл.
СТ-4/09	Белова О.Г.	Базы данных	хор.
СТ-4/09	Белова О.Г.	Базы данных	отл.
СТ-4/09	Белова О.Г.	Сетевые технологии	удовл.
СТ-4/09	Белова О.Г.	Сетевые технологии	хор.
СТ-4/09	Белова О.Г.	Сетевые технологии	отл.

Разносхемные операции РА

Пример декартова произведения реальных таблиц:

```
select  *
        from  depart,  emp;
```

Если в части FROM указываются 2 и более таблицы, то СУБД по умолчанию строит их декартово произведение.

Другая разносхемная операция – соединение: селекция от декартова произведения.

Примеры.

1. Список отделов и их сотрудников:

```
select  *
        from  depart, emp
        where  emp.depno = depart.did;
```

2. Список проектов и их участников:

```
select  *
        from  project,  emp,  job
        where  emp.tabno = job.tabno
           and  job.pro = project.pro;
```

Применение операции соединения

Задание 1: вывести сотрудников с указанием ролей, которые они исполняют в проектах.

```
select  e.name,  j.rel
    from  emp  e,  job  j
    where  e.tabNo = j.tabNo;
```

Задание 2: вывести список проектов с указанием их руководителей.

```
select  p.title,  e.name
     from  emp  e,   job  j,   project  p
     where e.tabno = j.tabno
              and  j.pro = p.pro
              and  j.rel = 'руководитель';
```

Применение операции соединения

Задание 3: вывести список сотрудников с указанием количества проектов, в которых они участвуют.

```
select name, count(*)
    from emp, job
    where emp.tabno=job.tabno
    group by emp.tabno, emp.name;
```

Задание 4: вывести список проектов, в которых участвует более 5 сотрудников.

```
select p.title, count(*)
    from job j, project p
    where p.pro = j.pro
    group by p.pro, p.title
   having count(*) > 5;
```

Общий алгоритм выполнения операции *SELECT*

1. Выбор записей из указанной таблицы (***from***).
2. Проверка для каждой записи условия отбора (***where***).
3. Группировка полученных в результате отбора записей (***group by***) и вычисление для этих групп значений агрегирующих функций.
4. Выбор тех групп, которые удовлетворяют условию отбора групп (***having***).
5. Сортировка полученных записей в указанном порядке (***order by***).
6. Извлечение из полученных записей тех полей, которые заданы в списке вывода, и формирование результирующего отношения.

Если в части FROM указывается 2 и более таблицы, то приведенный алгоритм выполняется для декартова произведения этих таблиц.