**17. Оператор SELECT. Использование подзапросов. Операции объединения,**пересечения и разности таблиц.

#### Использование подзапросов

Очень удобным средством, позволяющим формулировать запросы более понятным образом, является возможность использования подзапросов, вложенных в основной запрос.

**Пример**. Получить список людей, чья зарплата ниже среднего:

**SELECT** a.fam, a.zap

**FROM** a

**WHERE** a.zap < (**SELECT** **AVG**(a.zap) **FROM** a)

Замечание. Т.к. поле a.zap сравнивается с результатом подзапроса, то подзапрос должен быть сформулирован так, чтобы возвращать таблицу, состоящую *ровно из одной строки и одной колонки*.

Замечание. Результат выполнения запроса будет эквивалентен результату следующей последовательности действий:

1. Выполнить *один раз* вложенный подзапрос и получить среднее значение статуса.
2. Просканировать таблицу людей, каждый раз сравнивая значение средней зарплаты работника с результатом подзапроса, и отобрать только те строки, в которых зарплата меньше среднего.

**Пример**. Использование предиката ***EXIST***. Получить список людей, чья зарплата меньше, чем оклад:

**SELECT** a.fam, a.zap

**FROM** a

**WHERE** **EXISTS** (**SELECT** \*

**FROM** dolgnost

**WHERE** a.zap<dolgnost.money)

Замечание. Результат выполнения запроса будет эквивалентен следующей последовательности действий:

1. Просканировать таблицу А, *каждый раз выполняя подзапрос* с новым значением размера зарплаты, взятым из таблицы P.
2. В результат запроса включить только те строки из таблицы A, для которых вложенный подзапрос вернул непустое множество строк.

Замечание. Вложенный подзапрос содержит параметр (внешнюю ссылку), передаваемый из основного запроса – размер зарплаты a.zap. Такие подзапросы называются ***коррелируемыми*** (***correlated***). Внешняя ссылка может принимать различные значения для каждой строки-кандидата, оцениваемого с помощью подзапроса, поэтому подзапрос должен выполняться заново для каждой строки, отбираемой в основном запросе. Такие подзапросы характерны для предиката EXIST, но могут быть использованы и в других подзапросах.

Замечание. Может показаться, что запросы, содержащие коррелируемые подзапросы будут выполняться медленнее, чем запросы с некоррелируемыми подзапросами. На самом деле это не так, т.к. как пользователь, сформулировал запрос, и *не определяет*, как этот запрос будет выполняться. Язык SQL является непроцедурным, а декларативным. Это значит, что пользователь, формулирующий запрос, просто описывает, *каким должен быть результат запроса*, а как этот результат будет получен - за это отвечает сама СУБД.

**Пример 28**. Использование предиката ***NOT EXIST***. Получить список поставщиков, не поставляющих деталь номер 2:

SELECT \*

FROM P

WHERE NOT EXIST

(SELECT \*

FROM PD

WHERE

PD.PNUM = P.PNUM AND

PD.DNUM = 2);

Замечание. Также как и в предыдущем примере, здесь используется коррелируемый подзапрос. Отличие в том, что в основном запросе будут отобраны те строки из таблицы поставщиков, для которых вложенный подзапрос не выдаст ни одной строки.

**Пример 29**. Получить имена поставщиков, поставляющих все детали:

SELECT DISTINCT PNAME

FROM P

WHERE NOT EXIST

(SELECT \*

FROM D

WHERE NOT EXIST

(SELECT \*

FROM PD

WHERE

PD.DNUM = D.DNUM AND

PD.PNUM = P.PNUM));

Замечание. Данный запрос содержит два вложенных подзапроса и реализует реляционную операцию *деления отношений*.

Самый внутренний подзапрос параметризован двумя параметрами (D.DNUM, P.PNUM) и имеет следующий смысл: отобрать все строки, содержащие данные о поставках поставщика с номером PNUM детали с номером DNUM. Отрицание NOT EXIST говорит о том, что данный поставщик не поставляет данную деталь. Внешний к нему подзапрос, сам являющийся вложенным и параметризованным параметром P.PNUM, имеет смысл: отобрать список деталей, которые не поставляются поставщиком PNUM. Отрицание NOT EXIST говорит о том, что для поставщика с номером PNUM не должно быть деталей, которые не поставлялись бы этим поставщиком. Это в точности означает, что во внешнем запросе отбираются только поставщики, поставляющие все детали.

#### Использование объединения, пересечения и разности

NATURAL JOIN - Естественное соединение производится по всем столбцам таблиц А и В, имеющим одинаковые имена. В результатирующую таблицу одинаковые столбцы вставляются только один раз.

JOIN … ON - Соединение посредством предиката соединяет строки таблиц А и В посредством указанного предиката.

JOIN … USING - Соединение посредством имен столбцов соединяет отношения подобно естественному соединению по тем общим столбцам таблиц А и Б, которые указаны в списке USING.

OUTER - Ключевое слово OUTER (внешний) не является обязательными, оно не используется ни в каких операциях с данными.

INNER - Тип соединения "внутреннее". Внутренний тип соединения используется по умолчанию, когда тип явно не задан. В таблицах А и В соединяются только те строки, для которых найдено совпадение.

LEFT (OUTER) - Тип соединения "левое (внешнее)". Левое соединение таблиц А и В включает в себя все строки из левой таблицы А и те строки из правой таблицы В, для которых обнаружено совпадение. Для строк из таблицы А, для которых не найдено соответствия в таблице В, в столбцы, извлекаемые из таблицы В, заносятся значения NULL.

RIGHT (OUTER) - Тип соединения "правое (внешнее)". Правое соединение таблиц А и В включает в себя все строки из правой таблицы В и те строки из левой таблицы А, для которых обнаружено совпадение. Для строк из таблицы В, для которых не найдено соответствия в таблице А, в столбцы, извлекаемые из таблицы А заносятся значения NULL.

FULL (OUTER) - Тип соединения "полное (внешнее)". Это комбинация левого и правого соединений. В полное соединение включаются все строки из обеих таблиц. Для совпадающих строк поля заполняются реальными значениями, для несовпадающих строк поля заполняются в соответствии с правилами левого и правого соединений.

UNION JOIN - Соединение объединения является обратным по отношению к внутреннему соединению. Оно включает только те строки из таблиц А и В, для которых не найдено совпадений. В них используются значения NULL для столбцов, полученных из другой таблицы. Если взять полное внешнее соединение и удалить из него строки, полученные в результате внутреннего соединения, то получится соединение объединения.

Использование соединенных таблиц часто облегчает восприятие оператора SELECT, особенно, когда используется естественное соединение. Если не использовать соединенные таблицы, то при выборе данных из нескольких таблиц необходимо явно указывать условия соединения в разделе WHERE. Если при этом пользователь указывает сложные критерии отбора строк, то в разделе WHERE смешиваются семантически различные понятия - как условия связи таблиц, так и условия отбора строк