

Лабораторная работа № 17.

ТЕМА: «Комбинированные действия с данными»

ЦЕЛЬ: Научиться делать комбинированные действия.

Порядок выполнения работы:

1. Воспользоваться предикатом Exists;
2. Воспользоваться предикатом in;
3. Воспользоваться подзапросом.

Теоретические сведения

Предикат exists

Синтаксис:

1. EXISTS::=
2. [NOT] EXISTS (<табличный подзапрос>)

Предикат **EXISTS** принимает значение **TRUE**, если подзапрос содержит любое количество строк, иначе его значение равно **FALSE**. Для **NOT EXISTS** все наоборот. Этот предикат никогда не принимает значение **UNKNOWN**.

Обычно предикат **EXISTS** используется в зависимых (коррелирующих) подзапросах. Этот вид подзапроса имеет внешнюю ссылку, связанную со значением в основном запросе. Результат подзапроса может зависеть от этого значения и должен оцениваться отдельно для каждой строки запроса, в котором содержится данный подзапрос. Поэтому предикат **EXISTS** может иметь разные значения для разных строк основного запроса.

Пример на пересечение.

Найти тех производителей портативных компьютеров, которые также производят принтеры:

1. SELECT DISTINCT maker
2. FROM Product AS lap_product
3. WHERE type = 'laptop' AND
4. EXISTS (SELECT maker
5. FROM Product
6. WHERE type = 'printer' AND
7. maker = lap_product.maker
8.);

В подзапросе выбираются производители принтеров и сравниваются с производителем, значение которого передается из основного запроса. В основном же запросе отбираются производители портативных компьютеров. Таким образом, для каждого производителя портативных компьютеров проверяется, возвращает ли подзапрос строки (которые говорят о том, что этот производитель также выпускает принтеры). Поскольку два условия в предложении **WHERE** должны выполняться одновременно (**AND**), то в результирующий набор попадут нужные нам строки.

DISTINCT используется для того, чтобы каждый производитель присутствовал в выходных данных только один раз.

Пример на разность.

Найти производителей портативных компьютеров, которые не производят принтеров:

1. SELECT DISTINCT maker
2. FROM Product AS lap_product
3. WHERE type = 'laptop' AND
4. NOT EXISTS (SELECT maker
5. FROM Product
6. WHERE type = 'printer' AND
7. maker = lap_product.maker
8.);

В этом случае достаточно заменить в предыдущем примере **EXISTS** на **NOT EXISTS**. То есть выходные данные составят только те уникальные строки основного запроса, для которых подзапрос не возвращает ни одной строки.

Предикат **in**.

Синтаксис:

1. **IN**::=
2. <Проверяемое выражение> [**NOT**] **IN** (<подзапрос>)
3. | (<выражение для вычисления значения>,...)

Предикат **IN** определяет, будет ли значение проверяемого выражения обнаружено в наборе значений, который либо явно определен, либо получен с помощью табличного подзапроса. Здесь табличный подзапрос это обычный оператор **SELECT**, который создает одну или несколько строк для одного столбца, совместимого по типу данных со значением проверяемого выражения. Если целевой объект эквивалентен хотя бы одному из указанных в предложении **IN** значений, истинностное значение предиката **IN** будет равно **TRUE**. Если для каждого значения **X** в предложении **IN** целевой объект $\langle \rangle$ **X**, истинностное значение будет равно **FALSE**. Если подзапрос выполняется, и результат не содержит ни одной строки (пустая таблица), предикат принимает значение **FALSE**. Когда не соблюдается ни одно из упомянутых выше условий, значение предиката равно **UNKNOWN**.

Пример 5.2.4

Найти модель, частоту процессора и объем жесткого диска тех компьютеров, которые комплектуются накопителями 10 или 20 Гбайт:

1. SELECT model, speed, hd
2. FROM PC
3. WHERE hd IN (10, 20);

Пример 5.2.5

Найти модель, частоту процессора и объем жесткого диска компьютеров, которые комплектуются накопителями 10 Гбайт или 20 Гбайт и выпускаются производителем A:

1. SELECT model, speed, hd
2. FROM PC
3. WHERE hd IN (10, 20) AND
4. model IN (SELECT model
5. FROM product
6. WHERE maker = 'A'
7.);

Подзапрос

Заметим, что в общем случае запрос возвращает множество значений. Поэтому использование подзапроса в предложении **WHERE** без предикатов **EXISTS**, **IN**, **ALL** и **ANY**, которые дают булево значение, может привести к ошибке времени выполнения запроса.

Пример 5.8.3

Найти модели и цены ПК, стоимость которых превышает минимальную стоимость портативных компьютеров:

1. SELECT DISTINCT model, price
2. FROM PC
3. WHERE price > (SELECT MIN(price)
4. FROM Laptop
5.);

Этот запрос вполне корректен, так как скалярное значение price сравнивается с подзапросом, который возвращает единственное значение. В результате получим три модели ПК:

Однако, если в ответ на вопрос «найти модели и цены ПК, стоимость которых совпадает со стоимостью портативных компьютеров» написать следующий запрос

1. SELECT DISTINCT model, price
2. FROM PC
3. WHERE price = (SELECT price
4. FROM Laptop
5.);

то при выполнении последнего мы можем получить такое сообщение об ошибке: *Subquery returned more than 1 value. This is not permitted when the subquery follows =, !=, <, <=, >, >= or when the subquery is used as an expression.*

(«Подзапрос вернул более одного значения. Это не допускается в тех случаях, когда подзапрос следует после =, !=, <, <=, >, >= или когда подзапрос используется в качестве выражения».)

Эта ошибка будет возникать при сравнении скалярного значения с подзапросом, который возвращает более одного значения.

Подзапросы, в свою очередь, также могут содержать вложенные запросы.

С другой стороны, подзапрос, возвращающий множество строк и содержащий несколько столбцов, вполне естественно может использоваться в предложении **FROM**. Это, например, позволяет ограничить набор столбцов и/или строк при выполнении операции соединения таблиц.

Пример 5.8.4

Вывести производителя, тип, модель и частоту процессора для Портативных компьютеров, частота процессора которых превышает 600 МГц.

Этот запрос может быть сформулирован, например, следующим образом:

1. SELECT prod.maker, lap.*
2. FROM (SELECT 'laptop' AS type, model, speed
3. FROM laptop
4. WHERE speed > 600
5.) AS lap INNER JOIN
6. (SELECT maker, model
7. FROM product
8.) AS prod ON lap.model = prod.model;

Наконец, подзапросы могут присутствовать в предложении **SELECT**. Это иногда позволяет весьма компактно сформулировать запрос.

Пример 5.8.5

Найти разницу между средними значениями цены портативных компьютеров и ПК, то есть насколько в среднем портативный компьютер стоит дороже, чем ПК.

Здесь вообще можно обойтись одним предложением **SELECT** в основном запросе:

1. SELECT (SELECT AVG(price)
2. FROM Laptop
3.) -
4. (SELECT AVG(price)
5. FROM PC
6.) AS dif_price;

Краткая информация о базе данных "Компьютерная фирма":

Схема БД состоит из четырех таблиц:

Product(maker, model, type)

PC(code, model, speed, ram, hd, cd, price)

Laptop(code, model, speed, ram, hd, price, screen)

Printer(code, model, color, type, price)

Таблица Product представляет производителя (maker), номер модели (model) и тип ('PC' - ПК, 'Laptop' - ПК-блокнот или 'Printer' - принтер). Предполагается, что номера моделей в таблице Product уникальны для всех производителей и типов продуктов. В таблице PC для каждого ПК, однозначно определяемого уникальным кодом – code, указаны модель – model (внешний ключ к таблице Product), скорость - speed (процессора в мегагерцах), объем памяти - ram (в мегабайтах), размер диска - hd (в гигабайтах), скорость считывающего устройства - cd (например, '4x') и цена - price.

Таблица Laptop аналогична таблице PC за исключением того, что вместо скорости CD содержится размер экрана -screen (в дюймах). В таблице Printer для каждой модели принтера указывается, является ли он цветным - color ('y', если цветной), тип принтера - type (лазерный – 'Laser', струйный – 'Jet' или матричный – 'Matrix') и цена - price.

Задание:

- 1) Найдите средний размер диска ПК (одно значение для всех) тех производителей, которые выпускают и принтеры. Вывести: средний размер HD
- 2) Найдите производителей принтеров, которые производят ПК с наименьшим объемом RAM и с самым быстрым процессором среди всех ПК, имеющих наименьший объем RAM. Вывести: Maker
- 3) Найдите среднюю цену ПК и ПК-блокнотов, выпущенных производителем A (латинская буква). Вывести: одна общая средняя цена.

Ход работы:

1. Прочитать теорию.
2. Ознакомиться с базой данных "Компьютерная фирма".
3. Выполнить задание.

Контрольные вопросы:

1. Описать формат предиката exists?
2. Описать формат предиката in?
3. Описать подзапрос?