**Лабораторная работа № 4 SQL запросы на соединение таблиц (СХЕМА HR)**  
  
**Цель работы:** изучить методы выборки данных из нескольких таблиц и получить начальные навыки организации многотабличных запросов.  
  
**Декартово произведение**– получение результирующего отношения при присоединении к каждой строке одной таблицы каждой строки другой таблицы. 

|  |
| --- |
| ***Пример:*** получить выборку данных из таблиц EMPLOYEES, DEPARTMENTS. SQL> select last\_name, department\_name from employees, departments; |

В результирующем отношении данные никак логически не связаны. Чтобы увязать их, необходимо указать способ соединения таблиц.   
  
**Соединение** создает декартовое произведение двух таблиц и из полученного результирующего отношения выбирает данные на основе предиката.  
  
**Эквисоединения**– соединения, в которых используются предикаты с равенствами.  
  
**Естественное эквисоединение соединение**– соединение, основанное на равенстве значений внешнего ключа дочерней таблицы и первичного ключа родительской таблицы.   
  
**«Неестественное» эквисоединение соединение**– соединение, основанное на равенстве значений не ключевых полей таблиц.

|  |
| --- |
| ***Пример:*** вывести название отдела, в котором работает сотрудник по фамилии Fay (эквисоединение):  SQL> select a.last\_name, s.department\_name from employees a, departments s  2 where a.department\_id=s.department\_id and last\_name='Fay'  3 ; |

***Пример:*** вывести список сотрудников с указанием, в какую категорию по уровню зарплаты он попадает (не эквисоединение):  
  
SQL> select a.last\_name, a.salary, s.job\_id from employees a, jobs s

2 where a.salary between s.min\_salary and s.max\_salary;

SQL> select a.last\_name, a.salary, s.min\_salary, s.max\_salary from employees a, jobs s

2 where a.salary between s.min\_salary and s.min\_salary+0.01\*s.min\_salary;

***Пример:*** вывести список сотрудников с указанием фамилии их менеджера (соединение таблицы с собой):  
  
SQL> select a.last\_name||'---manager---'||m.last\_name

2 from employees a, employees m where

3 a.manager\_id=m.employee\_id order by a.manager\_id;

**Внешнее соединение** – расширение естественного соединения, включающего все строки из обоих отношений независимо от того, есть ли каждому кортежу из одного отношения соответствующий кортеж из другого отношения.  
  
Различают левое внешнее соединение (LEFT OUTER JOIN), правое внешнее соединении (RIGHT OUTER JOIN) и полное внешнее соединение (FULL OUTER JOIN).  
  
Таблица в соединении называется левой, если она стоит перед ключевым словом (JOIN), и правой, если она стоит после него.

|  |
| --- |
| ***Пример:*** вывести список всех сотрудников независимо от того числятся ли они в каком-либо отделе или нет, указать номера отделов и названий отделов (левое внешнее соединение) SQL> select w.last\_name as Employee, m.department\_id, m.department\_name  2 from employees w LEFT OUTER JOIN departments m  3 on (w.department\_id=m.department\_id); |

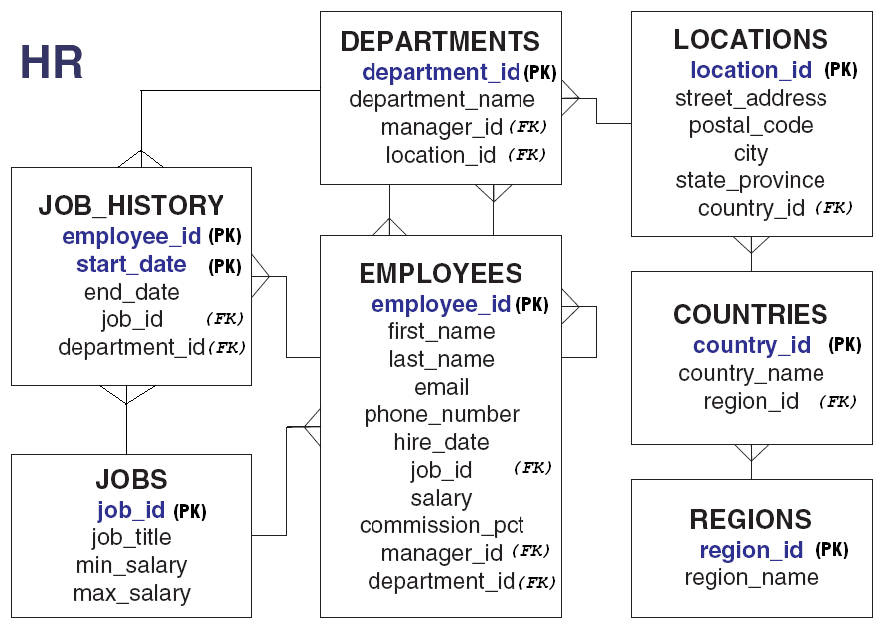
|  |
| --- |
| ***Пример:*** правое внешнее соединение  SQL> select w.last\_name as Employee, m.department\_id, m.department\_name  2 from employees w RIGHT OUTER JOIN departments m  3 on (w.department\_id=m.department\_id); |

В данном запросе возвращаются все строки таблицы departments, даже если они не соответствуют строкам таблицы employees. 

|  |
| --- |
| ***Пример:*** полное внешнее соединение  SQL> select w.last\_name as Employee, m.department\_id, m.department\_name  2 from employees w FULL OUTER JOIN departments m  3 on (w.department\_id=m.department\_id); |

В этом запросе возвращаются все строки таблицы employees, даже если они не соответствуют строкам таблицы departments, и наоборот.

|  |
| --- |
| **Создание трехсторонних соединений при помощи предложения JOIN … ON** SQL> select \* from  2 jobs j JOIN employees e ON j.job\_id=e.job\_id  3 JOIN departments d ON e.department\_id=d.department\_id; |



**Выполните следующие запросы и сохраните их в текстовом файле.**  
Настройка SQL. Необходимо включить режим ECHO и вывести протокол лаб. работы в файл *<Фамилия студента>.txt*. Этот файл является отчетом о проделанной лаб. работе.

SQL> set echo on

SQL> spool c:\spool. txt

……………………………………..

Завершить протокол лаб. работы (команда spool off) и направить результаты преподавателю.

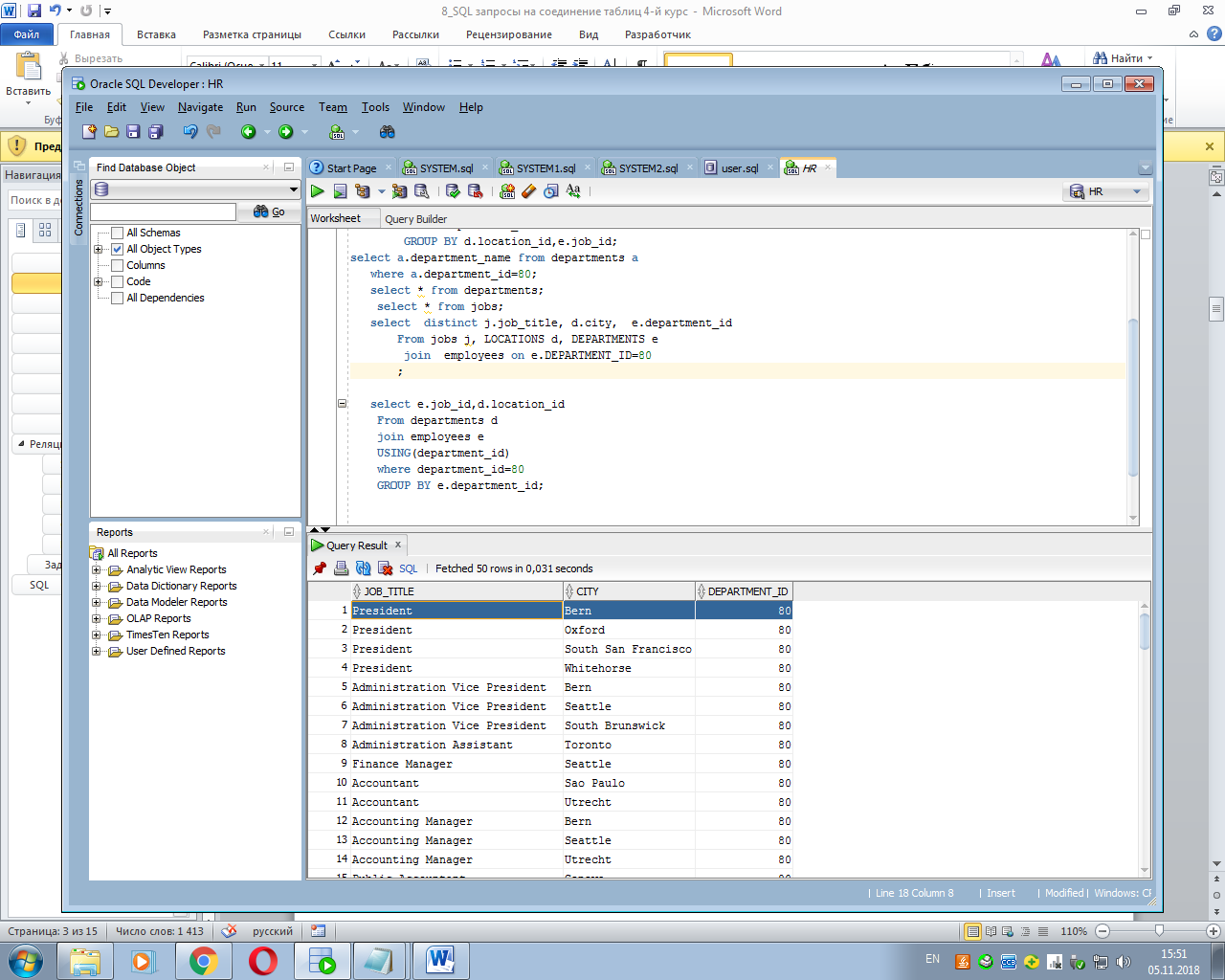
SQL> spool off

1. Выведите список всех должностей в отделе 80 (должности в списке не должны повторяться (DISTINCT)) и местоположение.
2. Вывести фамилии, названия отдела, город и местоположение отделов, для всех служащих, зарабатывающих комиссионные.
3. Вывести фамилии служащих, содержащие буквы «а» (в строчном регистре), с названиями отделов.
4. Напишите запрос для вывода фамилии, должности, номера отдела и названия отдела всех служащих, работающих в городе Toronto.
5. Вывести фамилии и номера служащих вместе с фамилиями и номерами их менеджеров. Назовите столбцы Employee, Emp#, Manager, MGR#.
6. Измените запрос 5 так, чтобы получить фамилии всех служащих, включая Кинга, который не имеет менеджера. Упорядочьте результат по возрастанию номера служащего.
7. Создайте запрос для вывода номера отдела, фамилии служащего и фамилий всех служащих, работающих в одном отделе с данным служащим. Дайте столбцам соответствующие имена.
8. Создайте запрос для вывода фамилии, должности, названия отдела, оклада всех служащих.
9. Создайте запрос для вывода фамилий и дат найма всех служащих, нанятых после Davies. (запрос выполнить используя декартово произведение, с использованием данной фамилии).
10. По всем служащим, нанятым раньше своих менеджеров, выведите фамилии и даты найма самих служащих, а также фамилии и даты найма их менеджеров.

select distinct j.job\_title, d.city, e.department\_id

From jobs j, LOCATIONS d, DEPARTMENTS e

join employees on e.DEPARTMENT\_ID=80;



SQL> select e.job\_id,d.location\_id

2 From departments d

3 join employees e

4 USING(department\_id)

5 where department\_id=80

6 GROUP BY d.location\_id,e.job\_id;

JOB\_ID LOCATION\_ID

---------- -----------

SA\_MAN 2500

SA\_REP 2500

# 



# 

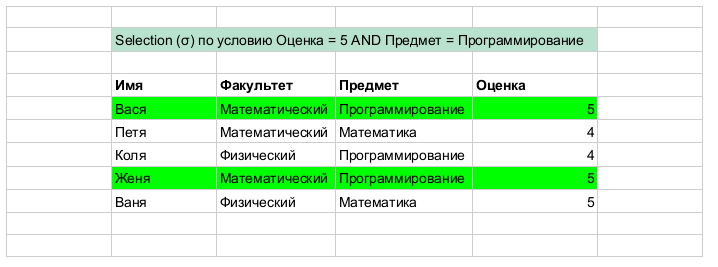
# Реляционная алгебра

Основные операторы

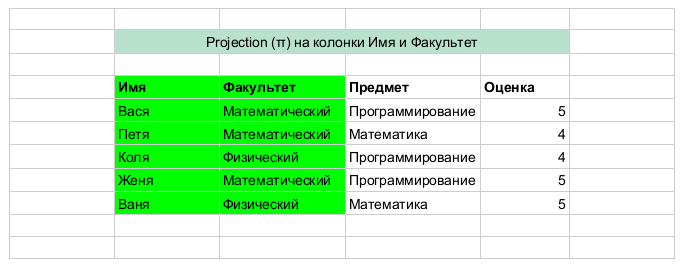
 — само отношение А (отношение здесь синонимично с таблицей и предикатом) является выражением реляционной алгебры, более того, так как это алгебра, любое выражение реляционной алгебры возвращает отношение (свойство замыкания операторов)

### Selection (выборка; ограничение)

 — selection (выборка; ограничение), A — отношение (предикат, таблица),  – булева формула, по которой происходит отбор строк (кортежей, записей, etc)

Selection является по сути горизонтальным фильтром строк, т.е., можно представить, что мы идем по каждой строке и оставляем только те, что удовлетворяют условию . Простой пример для наглядности:  


### Projection (проекция)

 — projection (проекция) на атрибуты A, B, …. Возвращает таблицу, в которой остаются только колонки (атрибуты) A, B, …. Простой пример ниже. По сути является фильтром по атрибутам т.е. это в некотором смысле вертикальный фильтр.  


### Переименование

 — переименовывает колонку a в b в отношении A (атрибут, аргумент предиката, etc); два чая тому господину, который покажет, что алгебра строго более выразима с оператором переименования (нужно привести пример запроса, который не выразим без этого оператора, но выразим с )

**Декартово произведение**

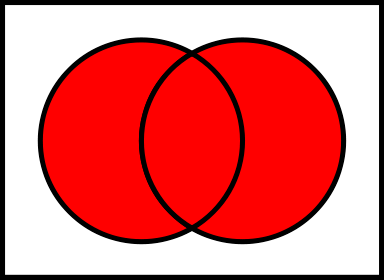
 — Декартово произведение двух отношений, большое отношение из всевозможных сочетаний строк в A и B.



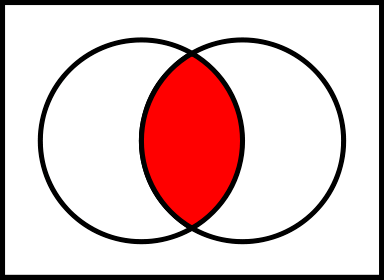
**Операции над множествами**

Реляционная алгебра является расширением классического набора операторов над множествами (отношение — это множество упорядоченных кортежей; заметьте, что это совсем не равно упорядоченному множеству кортежей). Пусть у нас есть таблица StudentMark(Name, Mark, Subject, Date) – тогда кортеж (Вася, 5, Информатика, 05.10.2010) является упорядоченным – сначала строка Name на первой (ок, или нулевой) позиции, целое число на второй, строка на третьей и дата на четвертой. При этом сами упорядоченные кортежи (Name, Mark, Subject, Date) не упорядочены «внутри» отношения.

### Объединение



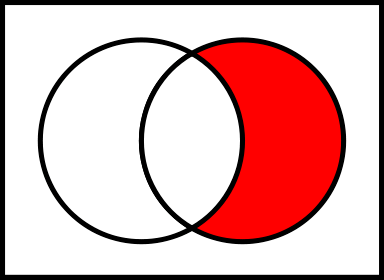
 — объединение всех строк в A и B, ограничение — одинаковые аттрибуты  
  
  
**Пересечение**

****

 — пересечение строк, такое же ограничение

### Разница множеств

 — B минус A, все строки, что присутствуют в B, но не присутствуют в A, такое же ограничение

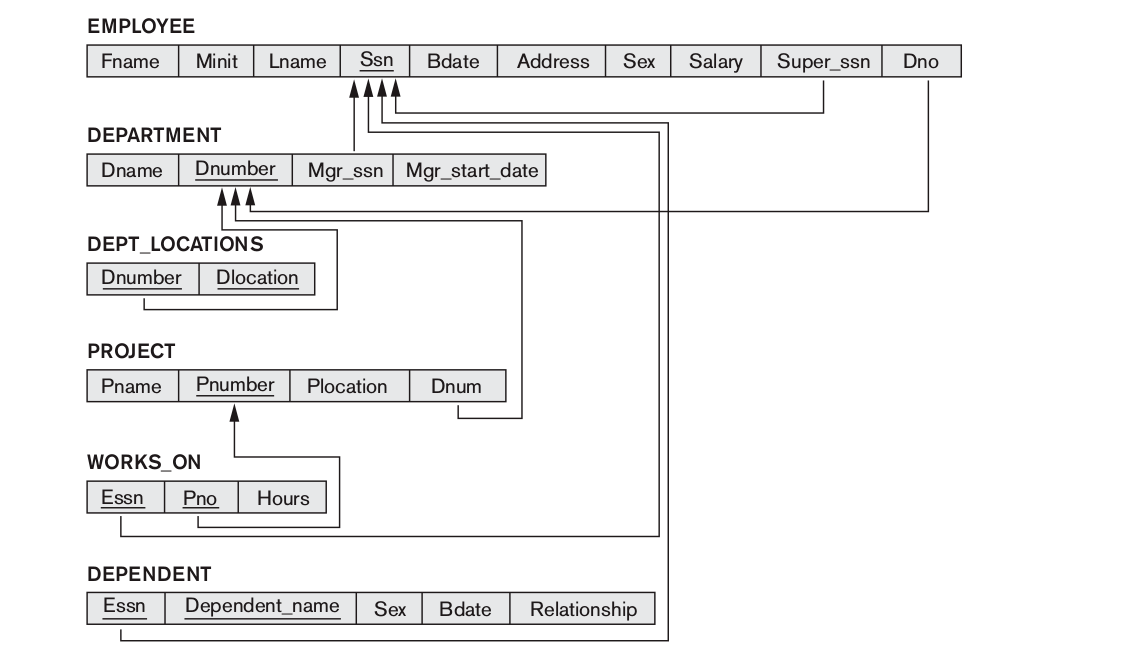


(B\A; A — слева, B — справа)

**Вспомогательные операторы**  
 — join (соединение); join соединяет две записи таблиц A и B, при условии, что для этих двух записей выполнено условие φ.



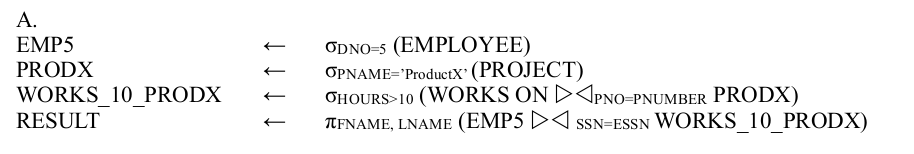
## Задачи для тренировки

Мы будем работать со следующей схемой  
  
  


**Задача первая.** Вывести имена всех работников 5го департамента, которые работают более 10 часов в неделю над проектом Х.

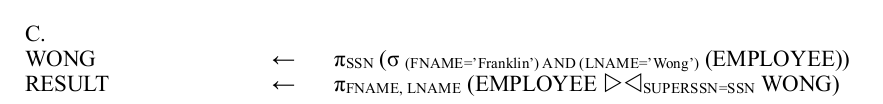
(Промежуточные результаты можно «сохранять» в новых отношениях, но это не обязательно.)

**Решение первой задачи. Реляционная алгебра**



**Задача вторая.** Вывести имена всех работников, которыми напрямую руководит Франклин Вонг (и найти небольшую ошибку в решении ниже)

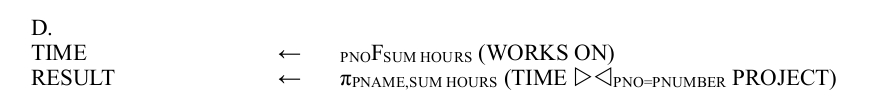
**Решение второй задачи. Реляционная алгебра**



**Задача третья** потребует использования нового оператора — «агрегация». Рассмотрим его на примере:

Для каждого проекта вывести название и общее число часов в неделю, которое все работники тратят на этот проект.

**Решение третьей задачи. Реляционная алгебра**



Заметим, что запрос имеет вид a F b (A), где a, b — колонки, A — отношение, а – агрегационная функция (например, SUM, MIN, MAX, COUNT, etc). Читается следующим образом: для каждого значения в колонке а, посчитай b. То есть одному значению в колонке a может соотвестовать несколько строк, поместим значения колонки b из этого множества строк в функцию и создадим новый атрибут fun\_b с соотвествующим значением.

Данный запрос не выразим в «классической» реляционной алгебре (без оператора агрегации F). То есть, нельзя написать единственный запрос, который бы для любой базы данных, удовлетворяющей схеме, выдавал бы правильный ответ.

Откуда в точности следует данные результат мы разберем позднее, сейчас можно лишь отметить, что запросы с агрегацией принадлежат более высокому классу вычислительной сложности.

Мы рассмотрим и проанализируем более интересные примеры задач далее в статье. Там же небольшая подборка задач на реляционную алгебру с решениями доступна [здесь](https://github.com/SergeyParamonov/HabraData/tree/master/Slides%20queries)

# SQL

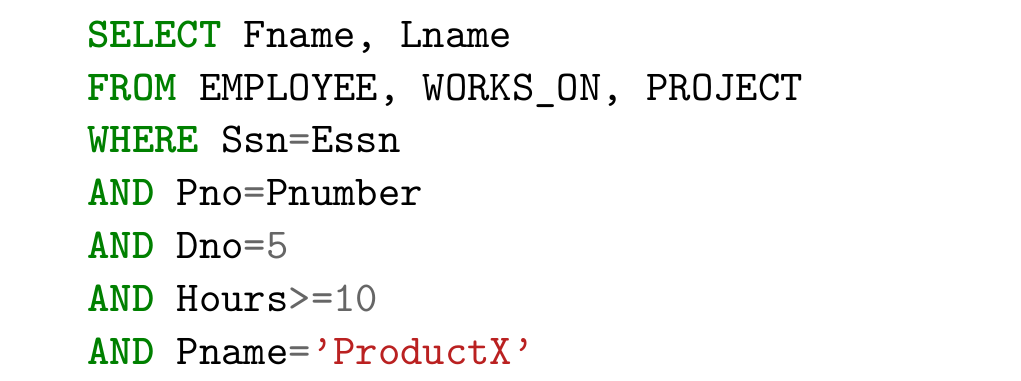
В данной части мы поговорим о SQL (Structured Query Language) и покажем, как SQL соответствует реляционной алгебре на простых примерах.

Рассмотрим самую первую задачу еще раз:

**Задача первая.** Вывести имена всех работников 5го департамента, которые работают более 10 часов в неделю над проектом Х.

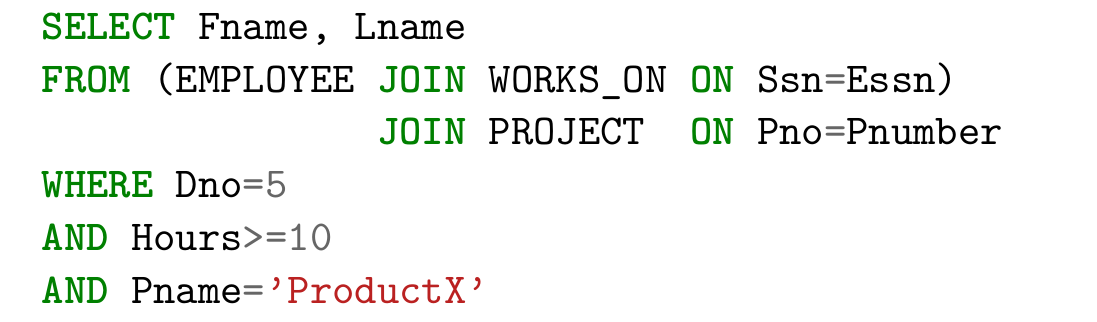
Классическое решение выглядит следующим образом:

**Классическое решение.**



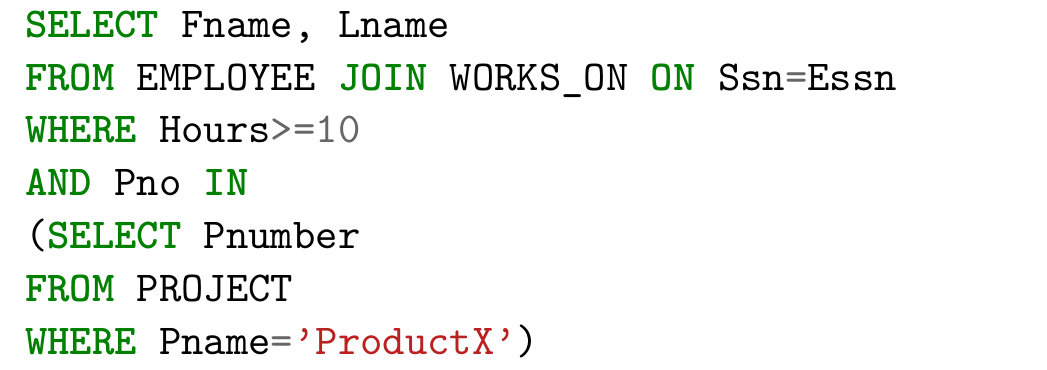
Альтернативно можно написать вот так:

**Немного альтернативно.**



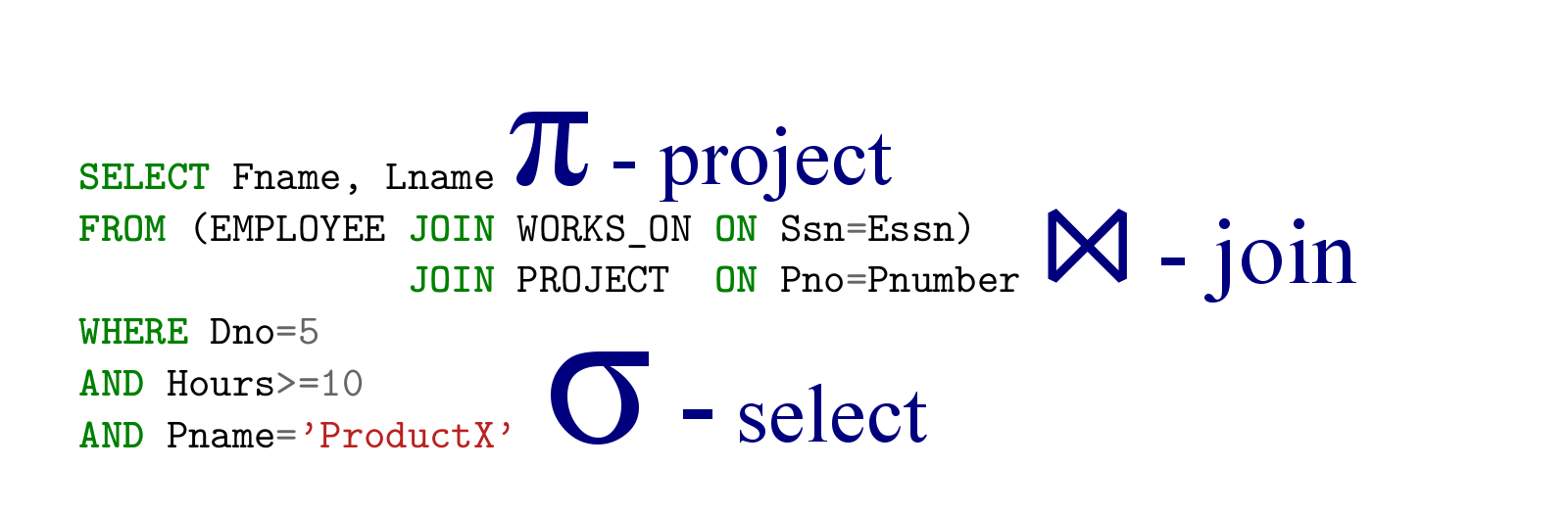
Или совсем альтернативно:

**С подзапросом**

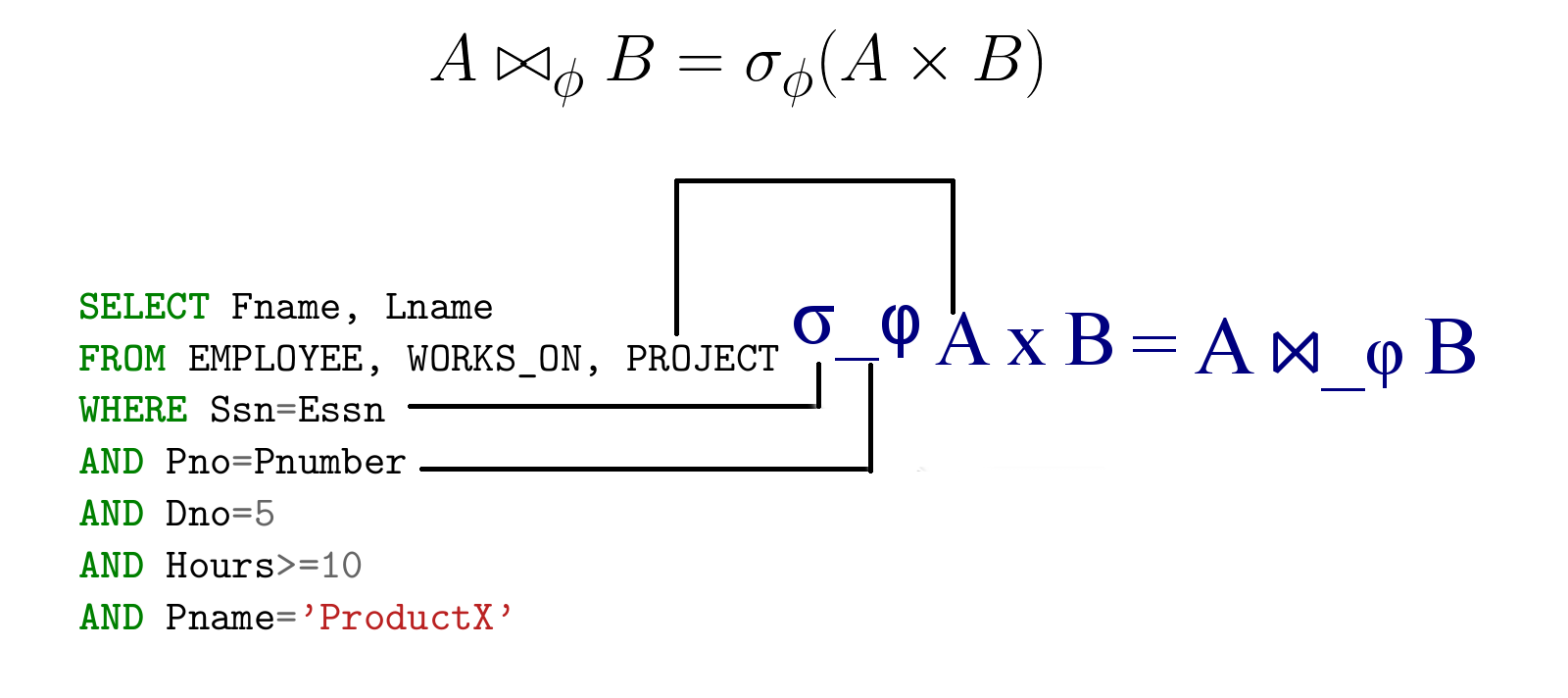


(далее решения не убраны под спойлеры)   
  
Проводим аналогию между SQL и реляционной алгеброй

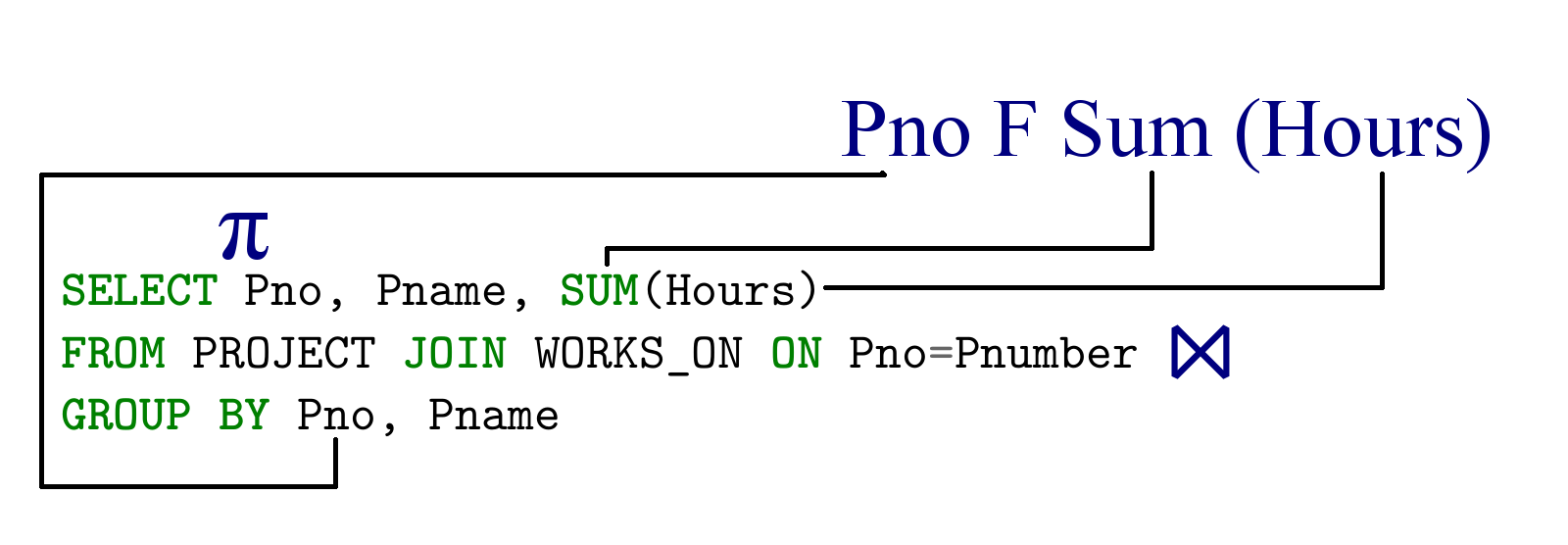
На втором решении мы видим отчетливую аналогию c реляционной алгеброй:



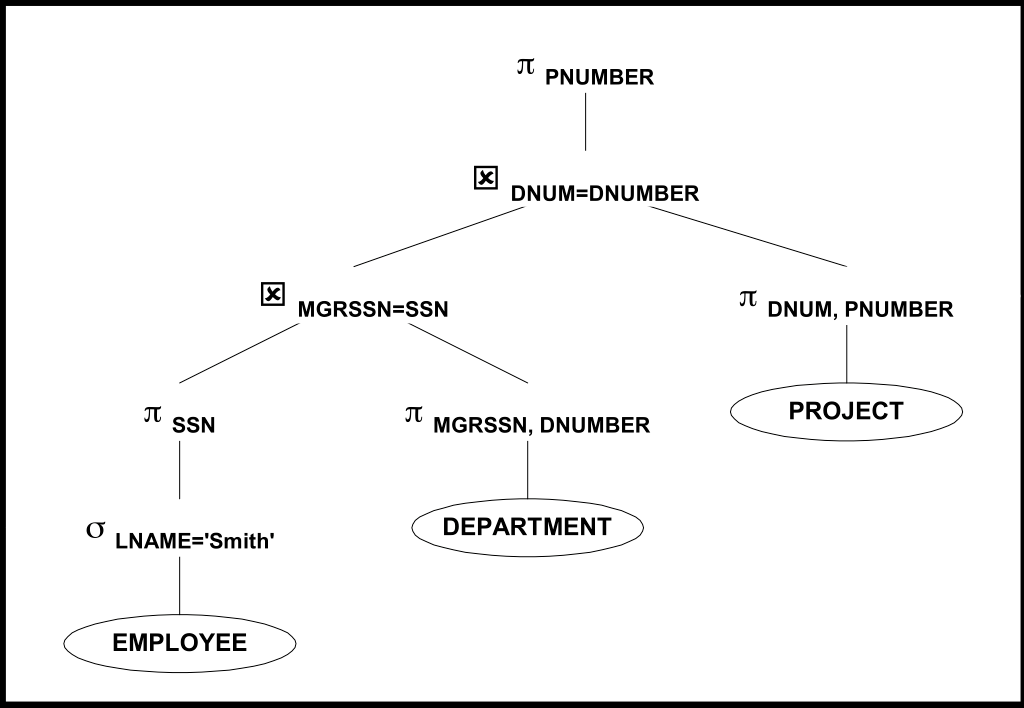
Теперь используем равенство для join и увидим аналогию между SQL и реляционной алгеброй в первом решении

  
Как бы это не было иронично, но SELECT в SQL — это project (π; проекция) в реляционной алгебре.

Теперь рассмотрим задачу с агрегацией и сравним её с решением на реляционной алгебре:



Более интересные задачи мы рассмотрим далее в статье (также небольшая подборка [здесь](https://dl.dropboxusercontent.com/u/12373089/Paramonov_SQL.pdf)), а сейчас рассмотрим еще один формализм запросов.

  
  
  
  
  
  
  
  
  
**Введите запрос**

**Введение в Oracle SQL**

**Функции преобразования типов данных Oracle:**

DECODE

(Правильный ответ) TO\_NUMBER

LOWER

(Правильный ответ) TO\_DATE

(Правильный ответ) TO\_CHAR

(Правильный ответ) TO\_CLOB

**Эффект, который оказывает на сетевой трафик использование хранимой процедуры, в которой выполняется несколько команд DML:**

большее количество повторных подключений к сети

большее количество полных обходов компьютерной сети

меньшее количество полных обходов компьютерной сети

(Правильный ответ) меньшее количество повторных подключений к сети

большее количество управляющих сигналов компьютерной сети

нет эффекта

**Во время выполнения транзакции произошла ошибка при выполнении команды DML. Вследствие этого будет произведен откат на уровне …**

выполнения

базы данных

(Правильный ответ) команды

системы

пользователя

**Пользователю «Alice» необходимо выполнить обновление данных таблицы VTABLE, при этом известно, что некоторые строки таблицы могут быть заблокированы транзакцией, открытой пользователем «Scott». Заблокировать свободные строки таблицы VTABLE с целью обновления позволит команда:**

SELECT \* FROM vtable FOR UPDATE NOWAIT;

SELECT \* FROM vtable LOCK TABLE;

SELECT \* FROM vtable FOR UPDATE OF Column2;

(Правильный ответ)

SELECT \* FROM vtable FOR UPDATE SKIP LOCKED;

**Команда запуска SQL\*Plus в командной строке операционной системы:**

RUN sq01/p01

(Правильный ответ) SQLPLUS sq01/p01

CONNECT sq01/p01

EXECUTE sq01/p01

**Предотвращение появления отношений с одинаковым набором столбцов в базе данных достигается методом …**

нормализации

централизации

(Правильный ответ) ортогонализации

денормализации

**При обновлении строк в таблице без условия WHERE произойдет следующая ситуация:**

будут обновлены только определенные строки таблицы

команда SQL не будет выполнена

(Правильный ответ) будут обновлены все строки таблицы

команда SQL будет выполнена, но не произойдет никаких изменений

Дана таблица:

VTABLE

A

B

C

4515261942122081041209

**Необходимо вычислить долю значения в столбце «b» в общей сумме значений столбца. Поставленную задачу выполнит запрос:**

(Правильный ответ)

SELECT a, b, c, RATIO\_TO\_REPORT (B) OVER () AS «»Analytic function»»FROM vtable;

SELECT a, b, c, SUM(c) OVER (ORDER BY с RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT\_ROW) AS «»Analytic function»»FROM vtable;

SELECT a, b, c, ROW\_NUMBER () OVER (ORDER BY a) AS «»Analytic function»»FROM vtable;

SELECT a, b, c, b — LAG(b,1) OVER (ORDER BY b) AS «»Analytic function»»FROM vtable;

**Транзакции в СУБД предназначены для …**

(Правильный ответ) поддержки целостности данных при выполнении множественных команд SQL

одновременного обновления всех данных в базе данных

отслеживания производительности базы данных и обнаружения медленно выполняемых команд

записи резервных копий базы данных

**Дана таблица S\_EMP, в которой содержится информация о сотрудниках компании:**

**IDНомер сотрудникаPRIMARY KEYLAST\_NAMEФамилияNOT NULLTITLEДолжностьNOT NULLMANAGER\_IDНомер начальника**

**SALARYЗарплатаNOT NULLCOMMISSION\_PCTКомиссионный процент**

**Запрос SQL:**

**SELECT id, last\_name, title,NVL2(commission\_pct,20,5) «»Комисионный процент»» FROM s\_emp;**

**В отчете, полученном при выполнении запроса, будут отображены следующие значения комиссионного процента различным категориям сотрудников:**

(Правильный ответ) 5% сотрудникам компании, ранее не получавшим комиссионные

5% всем сотрудникам компании

(Правильный ответ) 20% сотрудникам компании, ранее получавшим комиссионные

20% всем сотрудникам компании

**Дана таблица S\_EMP, в которой содержится информация о сотрудниках компании:**

**IDНомер сотрудникаPRIMARY KEYLAST\_NAMEФамилияNOT NULLTITLEДолжностьNOT NULLDEPT\_IDНомер отдела**

**SALARYЗарплатаNOT NULLSTART\_DATEДата начала работыNOT NULL**

**Необходимо получить фамилию, номер отдела, должность сотрудника с указанием суммарной зарплаты для каждой должности нарастающим итогом. Решением поставленной задачи является вариант:**

SELECT last\_name, title, dept\_id,SUM(salary) OVER (PARTITION BY dept\_id) sum\_sal FROM s\_emp;

SELECT last\_name, title, dept\_id, SUM(salary) sum\_sal FROM s\_empGROUP BY last\_name, title, dept\_id;

(Правильный ответ)

SELECT last\_name, title, dept\_id,SUM(salary) OVER (PARTITION BY title) sum\_sal FROM s\_emp;

SELECT last\_name, title, dept\_id,SUM(salary) OVER (PARTITION BY title,dept\_id) sum\_sal FROM s\_emp;

**В справочной базы данных компании имеется индекс, отсортированный в алфавитном порядке по фамилии. Такой тип индекса выгодно создавать, если большинство людей имеют что-либо из следующего:**

(Правильный ответ) разные и уникальные фамилии

пустые значения в столбцах фамилий

одинаковые фамилии

похожие фамилии

**Запросы со скалярными функциями аггрегирования:**

SELECT last\_name, title, dept\_id,SUM(salary) OVER (PARTITION BY title) sum\_sal FROM s\_emp;

SELECT id, last\_name, salary + NVL(commission\_pct,0) FROM s\_empWHERE dept\_id BETWEEN 31 AND 35ORDER BY last\_name;

SELECT last\_name, title, start\_date,AVG(salary) OVER (ORDER BY start\_date ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) avg\_sal FROM s\_emp;

(Правильный ответ)

SELECT manager\_id, MIN(salary) min\_sal, COUNT(id) emp\_countFROM s\_empWHERE manager\_id IS NOT NULLGROUP BY manager\_idHAVING min\_sal > 1000;

**Системный параметр NLS\_DATE\_FORMAT предназначен для настройки отображения …**

символов кириллицы

(Правильный ответ) формата даты

символа национальной валюты

кодировки

**Структуры SQL, которые допускают использование коллекций в качестве внутренних элементов:**

(Правильный ответ) строки таблиц

синонимы

ограничения

системные функции

псевдостолбцы

**В базе данных создан объектный тип RELATIVE\_TYPE, который предусматривает хранение имени, фамилии, даты рождения родственников. Команда SQL:**

**CREATE TYPE children\_list IS VARRAY(10) OF relative\_type;**

**Результат действия команды — создание …**

вложенной таблицы объектов

(Правильный ответ) массива объектов

таблицы объектов

тела объектного типа

**Дана таблица:**

**CUSTOMERS**

**ID**

**GENDER**

**DEPARTMENT**

**PURCHASES**

**1femalehousewares27.752malegarden42.203femalehome improvement97.504femalekitchen/bathroom28.605malefurniture225.756femalegarden34.407malehardware16.50**

**Запрос SQL:**

**SELECT department, COUNT(\*) FROM customers WHERE gender LIKE ‘%male’ AND purchases > 30.00 GROUP BY department ORDER BY 1;**

**Количество строк, возвращенных запросом, =…**

0

5

4

2

(Правильный ответ) 3

**Смысл рекурсивной связи, представленной на рисунке:**

В организации должны работать 0 или более начальников, наблюдающих за работой 1 или нескольких служащих

(Правильный ответ) У служащего может быть 0 или 1 начальник, и он может являться начальником у 0 или более служащих

Служащий может работать в одной или в нескольких организациях

В организации должны работать 0 или более служащих

0 или более служащих могут являться начальниками у 1 или более служащих

**Дана команда:**

**SELECT last\_name FROM s\_empORDER BY NLSSORT(last\_name, ‘NLS\_SORT = BINARY’);**

**Действие, выполняемое командой — установка порядка сортировки символьных строк …**

на основе двоичных кодов символов, с игнорированием регистра символов, для сеанса работы с сервером Oracle

на основе двоичных кодов символов, для сеанса работы с сервером Oracle

(Правильный ответ) на основе двоичных кодов символов, для результатов конкретного запроса

в соответствии с порядком латинского алфавита, результатов конкретного запроса

**Дана таблица S\_EMP, в которой содержится информация о сотрудниках компании:**

**IDНомер сотрудникаPRIMARY KEYLAST\_NAMEФамилияNOT NULLTITLEДолжностьNOT NULLMANAGER\_IDНомер начальника**

**SALARYЗарплатаNOT NULLSTART\_DATEДата начала работыNOT NULL**

**Запрос SQL:**

**SELECT last\_name, SUM(salary) OVER() sum\_sal FROM s\_emp;**

**Во избежание ошибки выполнения запроса для получения списка фамилий сотрудников и общей суммы зарплаты, в команду следует добавить:**

предложение PARTITION BY last\_name

предложение GROUP BY last\_name

(Правильный ответ) ничего

предложение PARTITION BY salary

**Даны таблицы:**

**ANIMALS**

**FAMILY**

**NAME**

**mammaldogmammallionfishguppybirdeaglefishguppybirdpelicanfishNULL**

**BLOOD\_TYPES**

**FAMILY**

**BLOOD\_TYPE**

**mammalwarm\_bloodedfishcold\_bloodedbirdcold\_blooded**

**Запрос SQL:**

**SELECT COUNT(DISTINCT name) FROM animals WHERE family = ANY (SELECT family FROM blood\_types WHERE blood\_type NOT LIKE (‘warm%’));**

**Количество строк, возвращенных запросом, = …**

(Правильный ответ) 1

5

4

3

2

**Запрос SQL:**

**WITH subordination (id, job) AS (SELECT id, title FROM s\_emp WHERE title = ‘President’UNION ALLSELECT s.job||’ -> ‘||e.titleFROM s\_emp e INNER JOIN subordination s ON s.id = e.manager\_id)SELECT job FROM subordination;**

**Основным запросом является выражение:**

SELECT id, title FROM s\_emp WHERE title = ‘President’

WITH subordination (id, job)

SELECT s.job||’ -> ‘||e.titleFROM s\_emp e INNER JOIN subordination s ON s.id = e.manager\_id

(Правильный ответ)

SELECT job FROM subordination

**Команда создания представления:**

**CREATE OR REPLACE VIEW emp\_info AS SELECT last\_name, first\_name, title, start\_date, dept\_id FROM employeeWHERE dept\_id = 30;**

**Запретить изменение таблицы EMPLOYEE через представление EMP\_INFO возможно при помощи опции …**

CONSTRAINT READ ONLY

WITH CHECK OPTION

(Правильный ответ) WITH READ ONLY

READ ONLY

**Запрос SQL:**

**1 SELECT bus\_name, profits 2 FROM business 3 WHERE city = 4 (SELECT city FROM locations 5 WHERE city LIKE ‘Alban%’ 6 AND state = ‘NY’) 7 ORDER BY profits;**

**Исключить вероятность возникновения ошибки, в случае возвращения подзапросом более одной строки, возможно изменением …**

строки 1 на: SELECT DISTINCT bus\_name, profits

строки 5 на: WHERE MAX(city) LIKE ‘Alban%’

строки 7 на: ORDER BY city, profits

строки 4 на: (SELECT DISTINCT city FROM locations

(Правильный ответ) строки 3 на: WHERE city = ANY

**Запрос SQL:**

**SELECT last\_name, title, dept\_id,SUM(salary) OVER (PARTITION BY title, dept\_id) sum\_sal FROM s\_emp;**

**Группами в данном запросе являются …**

множества строк с одинаковыми значениями в столбце TITLE

множества строк с одинаковыми значениями в столбце DEPT\_ID

все строки таблицы

(Правильный ответ) множества строк с одинаковыми комбинациями значений в столбцах TITLE и DEPT\_ID

**Запрос SQL**

**SELECT SYSDATE FROM DUAL;**

**возвращает …**

уникальный идентификатор пользователя базы данных, открывшего сеанс работы с сервером Oracle

уникальное имя пользователя базы данных, открывшего сеанс работы с сервером Oracle

дату и время суток, установленные на рабочей станции пользователя

(Правильный ответ) дату и время суток, установленные на сервере Oracle

**Верные объявления объектного типа в SQL Oracle:**

TYPE relative\_type IS RECORD (first\_name VARCHAR2(30),last\_name VARCHAR2(30),birth\_date DATE);

TYPE relative\_type IS TABLE OF relatives%ROWTYPE INDEX BY BINARY\_INTEGER;

CREATE TYPE BODY relative\_type AS MEMBER FUNCTION age RETURN NUMBER IS BEGIN RETURN TRUNC(SYSDATE — birth\_date); END; END;

(Правильный ответ)

CREATE TYPE relative\_type AS OBJECT (first\_name VARCHAR2(30),last\_name VARCHAR2(30),birth\_date DATE);

**Верные высказывания относительно аналитических функций на примере:**

SELECT SUM(…) OVER (PARTITION BY …) FROM …;

(Правильный ответ) множество строк таблицы разделено на подгруппы предложением PARTITION BY …

группой строк считаются все строки таблицы

функция возвращает одну строку и один аггрегированный результат

функция возвращает все строки таблицы и один аггрегированный результат для каждой строки

**Задать значения столбцов в команде INSERT в порядке, отличном от того, в котором следуют столбцы таблицы, позволяет …**

(Правильный ответ) список столбцов команды INSERT, например INSERT INTO employees (emp\_id, last\_name, first\_name)

предложение ORDER BY

команда ALTER TABLE

список значений команды INSERT, например INSERT INTO employees VALUES (24,’Smith’, ‘Barney’)

предложение SET

**Ключевое слово SAMPLE в команде SELECT используется для указания …**

символьного шаблона оператора LIKE

отмены значения метасимволов оператора LIKE

(Правильный ответ) пробной выборки строк таблицы

физического адреса строки таблицы

**Запрос SQL:**

**SELECT last\_name, title, start\_date,AVG(salary) OVER (ORDER BY start\_date RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) avg\_sal FROM s\_emp;**

**Команда иллюстрирует формирование интервалов аггрегирования по …**

строкам

условию

(Правильный ответ) значениям

группе

**При разработке структуры базы данных на ранней стадии исследования, в первую очередь, необходимо обратить внимание на …**

идентификацию ошибок

выбор технологии хранения

определение системных требований

(Правильный ответ) определение данных

определение операционных процессов

**Запрос SQL:**

**SELECT dept\_no, AVG(MONTHS\_BETWEEN(SYSDATE, hiredate)) FROM empWHERE AVG(MONTHS\_BETWEEN(SYSDATE, hiredate)) > 60GROUP BY dept\_no ORDER BY AVG(MONTHS\_BETWEEN(SYSDATE, hiredate));**

**Причина, вследствие которой запрос SQL вызывает ошибку:**

групповые функции не могут содержать вложенные однострочные функции

(Правильный ответ) групповые функции нельзя использовать в предложении WHERE

групповые функции нельзя использовать в предложении ORDER BY

групповые функции нельзя использовать в списке выборки команды SELECT

**Первое и последнее значения интервалов агрегирования позволяют получать функции:**

NULLS FIRST и NULLS LAST

(Правильный ответ) FIRST\_VALUE и LAST\_VALUE

UNBOUNDED и CURRENT ROW

ASCENDING и DESCENDING

**Запрос SQL:**

**SELECT a.ln, b.zip, c.city FROM a, b, c WHERE a.ln = b.ln;**

**Каждая из таблиц «a», «b» и «c» содержит по 100 строк; первичным ключом для таблиц «a» и «b» является столбец «ln». Максимальное количество строк, которое может возвратить данный запрос, = …**

100

100000

1

1000000

(Правильный ответ) 10000

**Верные высказывания относительно опций аналитических функций на примере:**

**SELECT SUM(…) OVER () FROM …;**

вывод пустых значений последними определяется предложением NULLS FIRST

разбиение множества строк таблицы на подгруппы выполняется предложением GROUP BY

отбор результирующих групп, удовлетворяющих условию, выполняется предложением HAVING

(Правильный ответ) порядок строк в подгруппах задается предложением ORDER BY

**Команда SQL, которая позволяет установить значение параметра сортировки символьных строк для сеанса работы с сервером Oracle:**

(Правильный ответ) ALTER SESSION

ALTER TABLE

ALTER USER

ALTER COLUMN

ALTER DATABASE

**Полный перечень узлов в структуре древовидной иерархии возвращает системная функция:**

CONNECT\_BY\_ROOT

CONNECT\_BY\_ISCYCLE

(Правильный ответ) SYS\_CONNECT\_BY\_PATH

CONNECT\_BY\_ISLEAF

**НЕ допустить изменений данных в таблице в SQL Oracle 11g позволит команда:**

UPDATE TABLE … READ ONLY;

(Правильный ответ) ALTER TABLE … READ ONLY;

FLASHBACK TABLE … READ ONLY;

TRUNCATE TABLE … READ ONLY;

**Условия, при которых следует создавать индекс:**

значения столбцов статичны и содержат ограниченный диапазон значений

таблица часто обновляется

таблица небольшого размера

(Правильный ответ) два столбца совместно используются в условии соединения

**Запрос SQL:**

**1 SELECT city, state, COUNT(\*) 2 FROM customers 3 GROUP BY city, state 4 UNION ALL 5 SELECT city, state, COUNT(\*) 6 FROM employees 7 GROUP BY city, state;**

**Необходимо отсортировать результат выборки, возвращаемой запросом. Для выполнения поставленной задачи следует разместить предложение ORDER BY …**

предложение ORDER BY не может быть включено в запрос, содержащий UNION ALL

между строками 3 и 4, и после строки 7

(Правильный ответ) после строки 7

после предложения UNION ALL

между строками 1 и 2, и между строками 5 и 6

**Привилегия REFERENCES разрешает пользователю создание …**

внешнего ключа в таблице другого пользователя

составного внешнего ключа

рекурсивного внешнего ключа

(Правильный ответ) внешнего ключа со ссылкой на таблицу другого пользователя

**При удалении строк из таблицы, с заданием тождественно ложного условия WHERE, произойдет следующая ситуация:**

команда SQL не будет выполнена

будут удалены все строки таблицы

(Правильный ответ) команда SQL будет выполнена, но не произойдет никаких изменений

будут удалены только определенные строки таблицы

**Текст последней команды SQL, выполненной в SQL\*Plus, содержится в …**

схеме пользователя

командном файле

схеме таблицы, к которой применялась команда

(Правильный ответ) буферном файле

**Cхема пользователя u01 в СУБД Oracle создается при выполнении команды:**

GRANT SCHEMA TO U01;

SELECT «u01» FROM DUAL;

ALTER USER u01 IDENTIFIED BY u1\_02p;

(Правильный ответ) CREATE USER u01 PROFILE «DEFAULT» IDENTIFIED BY u1\_02p;

**Команда SQL:**

**SELECT object\_name FROM all\_objectsWHERE object\_type = ‘TABLE’;**

**Результат действия команды — вывод имен всех …**

объектов, к которым пользователь имеет доступ

(Правильный ответ) таблиц, к которым пользователь имеет доступ

таблиц пользователя, чьим собственником он является

объектов пользователя, чьим собственником он является

**Дана команда:**

**ALTER SESSION SET NLS\_SORT = RUSSIAN\_CI;**

**Действие, выполняемое командой — установка порядка сортировки символьных строк …**

(Правильный ответ) в соответствии с правилами русского языка с игнорированием регистра символов, для сеанса работы с сервером Oracle

в соответствии с правилами русского языка, для результатов конкретного запроса

на основе двоичных кодов символов, для сеанса работы с сервером Oracle

в соответствии с правилами русского языка с учетом регистра символов, для сеанса работы с сервером Oracle

**Для корректного распределения информации и установки связи между родительской (главной) и дочерней (подчиненной) таблицами БД необходимо …**

создание первичного ключа в дочерней (подчиненной) таблице

(Правильный ответ) создание первичного ключа в родительской (главной) таблице

(Правильный ответ) создание внешнего ключа в дочерней (подчиненной) таблице

создание внешнего ключа в родительской (главной) таблице

**Ограничение, позволяющее обеспечить уникальность строк при создании таблицы базы данных:**

NOT NULL

(Правильный ответ) PRIMARY KEY

FOREIGN KEY

CHECK

**Для сортировки символьных строк в соответствии с правилами русского языка необходимо установить значение параметра NLS\_SORT равным …**

(Правильный ответ) RUSSIAN

LINGUISTIC

CYRILLIC

BINARY

RUS

Запрос SQL:

SELECT name, salary, dept\_no FROM employeeWHERE salary > (SELECT AVG(salary) FROM employee; WHERE dept\_no = (SELECT dept\_no FROM employee WHERE last\_name = (SELECT last\_name FROM employee WHERE salary > 50000)));

**Команда, результат действия которой будет вычислен в последнюю очередь:**

SELECT last\_name …

SELECT dept\_no …

SELECT AVG(salary) …

(Правильный ответ) SELECT name, salary, dept\_no …

**Групповая функция, возвращающая среднее арифметическое значение:**

(Правильный ответ) AVG

VARIANCE

MAX

MIN

SUM

**Опции PRECEDING и FOLLOWING аналитических функций определяют …**

порядок сортировки значений в группах

(Правильный ответ) границы аггрегирования

группы аггрегирования

порядок вывода пустых значений

**Дана таблица S\_EMP, в которой содержится информация о сотрудниках компании:**

**IDНомер сотрудникаPRIMARY KEYLAST\_NAMEФамилияNOT NULLTITLEДолжностьNOT NULLMANAGER\_IDНомер начальника**

**SALARYЗарплатаNOT NULLCOMMISSION\_PCTКомиссионный процент**

**Необходимо получить список всех сотрудников с указанием суммарного дохода, полученного сложением сумм зарплаты и комиссионного процента; при этом следует учесть, что у некоторых сотрудников значение комиссионого процента отсутствует. Запрос, который выполняет поставленную задачу:**

(Правильный ответ)

SELECT id, last\_name, salary + NVL(commission\_pct,0) «»Суммарный доход»»FROM s\_emp;

SELECT id, last\_name, NVL(salary + commission\_pct,0) «»Суммарный доход»»FROM s\_emp;

SELECT id, last\_name, salary + commission\_pct «»Суммарный доход»»FROM s\_emp;

SELECT id, last\_name, NVL(salary,0) + commission\_pct «»Суммарный доход»»FROM s\_emp;

**Совокупность объектов пользователя в СУБД называется …**

схема таблицы

группа данных

(Правильный ответ) схема пользователя

структура данных

**Псевдостолбец SQL Oracle 11g для получения данных из столбцов XMLTYPE:**

XMLSCHEMA

(Правильный ответ) XMLDATA

XMLAGG

XMLELEMENT

Таблицы базы данных:Запрос SQL:

SELECT \* FROM s\_region, s\_dept, s\_emp, s\_customer;

**Данные о количестве строк и столбцов таблиц:**

Имя таблицыЧисло столбцовЧисло строкS\_REGION25S\_DEPT310S\_EMP1325S\_CUSTOMER1210

**Выборка, возвращенная запросом, будет включать в себя …**

936 столбцов и 12500 строк

(Правильный ответ) 30 столбцов и 12500 строк

30 столбцов и 55 строк

28 столбцов и 2500 строк

**В базе данных существуют таблицы AUTHORS (авторы) и EMPLOYEES (сотрудники). Запросы SQL, возвращающие список фамилий (столбец NAME) всех авторов и сотрудников:**

Select NAME From AUTHORS Where NAME in (Select NAME From EMPLOYEES)

(Правильный ответ) Select NAME From AUTHORS union Select NAME From EMPLOYEES

Select distinct NAME From AUTHORS, EMPLOYEES;

Select NAME From AUTHORS inner join EMPLOYEES

**Для таблицы СУБД Oracle, соответствующей реляционной модели, верны утверждения:**

**(Правильный ответ) ни один столбец первичного ключа не может содержать NULL**

**(Правильный ответ) определитель NULL указывает, что значение столбца неопределено**

**ни один столбец внешнего ключа не может содержать NULL**

**определитель NULL — нулевое значение столбца**

**Предложение FROM запроса SQL:**

**FROM emp e, dept d**

**Верные варианты обращения к столбцам таблиц в предложении WHERE:**

(Правильный ответ) e.dept\_id = d.id

dept\_id = d.id

dept\_id = id

emp.dept\_id = dept.id

**Хранение в отдельном поле таблицы «внутренней» таблицы позволяет структура SQL Oracle 11g …**

OBJECT TABLE

(Правильный ответ) NESTED TABLE

INNER TABLE

PL/SQL TABLE

Дана таблица:

VTABLE

ID

A

B

C

1catdogmouse2liondogzebra3lambbirdferret4lynxantbat5flybeebeetle6fishsharkcricket

**Запрос SQL:**

**SELECT a, b FROM vtable WHERE c LIKE ‘%e%’;**

**Количество строк, возвращенных запросом, = …**

(Правильный ответ) 5

2

6

4

1

3

**Дана таблица S\_EMP, в которой содержится информация о сотрудниках компании:**

**IDНомер сотрудникаPRIMARY KEYLAST\_NAMEФамилияNOT NULLTITLEДолжностьNOT NULLMANAGER\_IDНомер начальника**

**SALARYЗарплатаNOT NULLSTART\_DATEДата начала работыNOT NULL**

**Необходимо вычислить количество месяцев между самой ранней и самой поздней датами приема на работу сотрудников. Запрос, который выполняет поставленную задачу:**

SELECT MONTH\_BETWEEN(MAX(start\_date), MIN(start\_date)) FROM s\_emp;

SELECT \* FROM s\_empWHERE start\_date BETWEEN MAX(start\_date) AND MIN(start\_date);

SELECT \* FROM s\_empWHERE MONTHS\_BETWEEN(MAX(start\_date), MIN(start\_date));

(Правильный ответ)

SELECT MONTHS\_BETWEEN(MAX(