**Проверочная работа 7**

1. Какой из запросов будет соответствовать сортировке такого вывода?

**Ответ 2**

|  |  |
| --- | --- |
| **col\_1** | **col\_2** |
| 1 | C |
| 1 | B |
| 1 | A |
| 2 | C |
| 2 | B |
| 2 | A |
| 3 | C |
| 3 | B |
| 3 | A |

Вариант №1:

|  |  |
| --- | --- |
|  | SELECT \*  FROM Table\_1  ORDER BY col\_1 DESC, col\_2 ASC |

Вариант №2:

|  |  |
| --- | --- |
|  | SELECT \*  FROM Table\_1  ORDER BY col\_1 ASC, col\_2 DESC |

Вариант №3:

|  |  |
| --- | --- |
|  | SELECT \*  FROM Table\_1  ORDER BY col\_1, col\_2 ASC |

1. Что будет в результате выполнения следующего скрипта?

**Ответ: Выведутся все данные из таблицы Employees, rollback отменит truncate**

BEGIN TRAN

TRUNCATE TABLE Employees

ROLLBACK

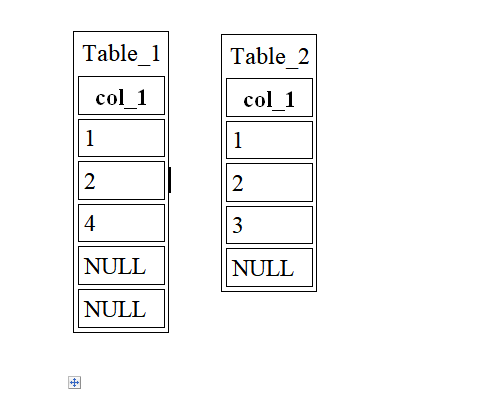
SELECT \* FROM Employees

3. Какой вывод соответствует запросу ниже с объединением JOIN?

В БД имеются 2 таблицы:

|  |
| --- |
| Table\_1 |
| **col\_1** |
| 1 |
| 2 |
| 4 |
| NULL |
| NULL |

|  |
| --- |
| Table\_2 |
| **col\_1** |
| 1 |
| 2 |
| 3 |
| NULL |



|  |  |
| --- | --- |
|  | SELECT \*  FROM Table\_1  JOIN Table\_2      ON Table\_1.col\_1 = Table\_2.col\_1 |

Какой вывод вернет база?

**Col 1 Col 1**

**1 1**

**2 2**

1. Какой вывод соответствует запросу ниже с объединением LEFT JOIN?

В БД имеются 2 таблицы:

|  |
| --- |
| Table\_1 |
| **col\_1** |
| 1 |
| 2 |
| 4 |
| NULL |
| NULL |

|  |
| --- |
| Table\_2 |
| **col\_1** |
| 1 |
| 2 |
| 3 |
| NULL |

К БД посылается такой запрос:

|  |  |
| --- | --- |
|  | SELECT \*  FROM Table\_1  LEFT JOIN Table\_2      ON Table\_1.col\_1 = Table\_2.col\_1 |

Какой вывод вернет база?

Col 1 Col 1

**1 1**

**2 2**

**4 NULL**

5. Какое значение будет возвращено в результате следующего запроса?

**Ответ 120**

SELECT AVG(TotalPayment)

FROM Orders

|  |  |
| --- | --- |
| OrderNo | TotalPayment |
|  |  |
| 1 | 100 |
|  |  |
| 2 | 200 |
|  |  |
| 3 | 100 |
|  |  |
| 4 | NULL |
|  |  |
| 5 | 100 |
|  |  |
| 6 | 100 |
|  |  |

1. Чем различаются запросы ниже?

Представленные запросы выполняются к одной и той же таблице одинакового наполнения. Таблица состоит из одного столбца Col\_1.

**Ответ: 1ый запрос посчитает количество всех строк, 2ой запрос — количество строк с уникальным значением**

Запрос №1:

|  |  |
| --- | --- |
|  | SELECT COUNT(\*) AS Cnt  FROM Table\_1 |

Запрос №2:

|  |  |
| --- | --- |
|  | SELECT COUNT(DISTINCT Col\_1) AS Cnt  FROM Table\_1 |

1. Какой тип отношения между таблицами «Покупатели» и «Заказы»?

**Ответ: один ко многим**

Имеется база данных интернет-магазина. В этой БД имеются таблицы:

* Покупатели – хранит информацию по клиентам (ФИО, логин, пароль и т.п.);
* Заказы – содержит номер заказа, комментарий, дату доставки, покупателя (который сделал заказ) и т.д.;
* Товары – id товара, наименование, цена, остатки и т.п.;

    Один к одному



    Один ко многим



    Многие ко многим



    Таблицы никак не связаны



1. Какой тип отношения между таблицами «Товары» и «Заказы»?

**Отает: многие ко многим**

Имеется база данных интернет-магазина. В этой БД имеются таблицы:

* Покупатели – хранит информацию по клиентам (ФИО, логин, пароль и т.п.);
* Заказы – содержит номер заказа, комментарий, дату доставки, покупателя (который сделал заказ) и т.д.;
* Товары – id товара, наименование, цена, остатки и т.п.;

    Один к одному



    Один ко многим



    Многие ко многим



    Таблицы никак не связаны



1. В какой нормальной форме находится база данных?

**Во второй нормлаьной форме**

В какой нормальной форме находится база данных?  
Имеется БД из одной таблицы «Сотрудники».  
Таблица содержит столбцы:

* Id – уникальный идентификатор сотрудника;
* ФИО – содержит фамилию, имя и отчество сотрудника;
* Дата рождения – дата рождения пользователя;
* Отдел\_id – идентификатор отдела, в котором пользователь работает;
* Отдел – наименование отдела, в котором пользователь работает.

    БД в первой нормальной форме



    БД во второй нормальной форме



    БД в третьей нормальной форме



    Ни одна нормальная форма не соблюдена



1. Какие строки вернет запрос c LIKE?

Имеется таблица:

|  |  |
| --- | --- |
| **ID** | **Value** |
| 1 | AAA |
| 2 | BBB |
| 3 | CCC |
| 4 | ABC |
| 5 | CBA |
| 6 | 123 |
| 7 | 213 |
| 8 | 321 |
| 9 | 1AB |
| 10 | BC3 |

К таблице выполняется запрос:

|  |  |
| --- | --- |
|  | SELECT \*  FROM Table\_1  WHERE id like '%A\_' |

Какие строки будут возвращены?

**Ответ: 1, 9**

1. Даны следующие таблицы:





Что будет результатом для следующего скрипта?

SELECT \* FROM runners

WHERE id NOT IN (SELECT winner\_id FROM races)

**Ответ: строки 1, 4, 5 из таблицы runners**

1. Даны следующие данные в колонке:

|  |
| --- |
| **x** |
| -2 |
| 2 |
| -5 |
| 0 |
| 1 |

Напишите скрипт для нахождения суммы всех положительных значений из колонки x или значений, кратных 5.

select sum(case when x<=0 then x  
 when x = 5 then 5  
 else 0   
 end)

from table