Вопросы:

Вопрос №1

Какой их этих запросов отсортирует записи по убыванию, а какой по возрастанию

a) ORDER BY "field" DESC – отсортирует записи по убыванию;
b) ORDER BY "field" ASC – отсортирует записи по возрастанию;

Вопрос №2

Какой из запросов выберет все записи, где значение поля "field" начинается с подстроки "abc"

a) SELECT * FROM `my_table` WHERE `field` LIKE '%abc%'
b) SELECT * FROM `my_table` WHERE `field` <> 'abc'
c) SELECT * FROM `my_table` WHERE `field` STARTSWITH 'abc'
Ответ: d) SELECT * FROM `my_table` WHERE `field` LIKE 'abc%'

Вопрос №3

Какой результат будет после выполнения запроса?

Select count(*) from table

1. Выберет все значения по всем столбцам

Ответ: 2) Количество записей в таблице

3. Количество столбцов в таблице

Вопрос №4

Что делает данный запрос?

select AAA, BBB, count(*) from test_table where CCC > 100 group by AAA, BBB

В данном запросе выбираются столбцы AAA и BBB из таблицы test_table, Оператор group by выполняет группировку по столбцам AAA и BBB, функция count(*) вычисляет количество строк в группе и после выполнения запроса добавляет в результат выборки столбец count(*), в котором после каждой группы число строк. В нашем запросе извлекаются не все данные из таблицы test_table, а только те, которые соответствуют определенному условию: поле CCC > 100 (значение поля CCC больше 100). Для фильтрации данных в команде select применяется оператор where.

Вопрос №5

Что получится в результате, после выполнения запроса:

*SELECT * FROM `my_table` WHERE 1=1*

1. Синтаксическая ошибка (из-за 1=1)

Ответ: 2) Выведутся все записи из таблицы "my_table".

3.Не выведется ни одной записи.

Вопрос №6

Для чего используется ключевое слово DISTINCT?

1. Для ускорения выборки по конкретному полю.
2. Для снижения нагрузки на сервер с потерей производительности выполнения запроса.

Ответ 3) Для выборки только уникальных записей по конкретному полю.

4.Такого ключевого слова не существует.

Вопрос №7

Имеем две таблицы:

U) users

id	name	d_id
1	Владимир	1
2	Антон	2
3	Александр	1
4	Борис	6

5	Юрий	4

D) departments

id	name
1	Сейлз
2	Поддержка
3	Финансы
4	Логистика

1) SELECT u.id, u.name, d.name AS d_name FROM users u **INNER JOIN** departments d ON u.d_id = d.id

Внутреннее объединение таблиц INNER JOIN

Запрос вернет объединенные данные, которые пересекаются по условию, указанному в INNER JOIN ON <..>.

В данном примере условие <таблица_пользователей>.<идентификатор_отдела> должен совпадать с <таблица_отделов>.<идентификатор>

Табліце users(таблица пользователей) присвоили псевдоним u, а таблице departments(таблица_отделов) псевдоним d. Select u.id означает выборку из таблицы u(пользователи) поля id. Select d.name означает выборку из таблицы d(отделы) поля id. Запись d.name AS d_name создание псевдонима для поля name таблицы d (departments)

2) SELECT u.id, u.name, d.name AS d_name FROM users u **Left OUTER JOIN** departments d ON u.d_id = d.id

Для получения данных, которые подходят по условию частично, в примере используют объединение - OUTER JOIN.

Такое объединение вернет данные полей u.id, u.name, d.name из обеих таблиц (совпадающие по условию объединения) ON u.d_id = d.id ПЛЮС дополнит выборку оставшимися данными из внешней таблицы, которые по условию не подходят, заполнив недостающие данные значением NULL.

3) SELECT u.id, u.name, d.name AS d_name FROM users u **Right OUTER JOIN** departments d ON u.d_id = d.id

Есть два типа внешнего объединения OUTER JOIN - LEFT OUTER JOIN и RIGHT OUTER JOIN.Работают они одинаково, разница заключается в том что LEFT - указывает что "внешней" таблицей будет находящаяся слева (в нашем примере это таблица users).

Ключевое слово OUTER можно опустить. Запись LEFT JOIN идентична LEFT OUTER JOIN.

RIGHT OUTER JOIN вернет полный список департаментов (правая таблица) и сопоставленных пользователей.

Объясните каждый из запросов.

Вопрос №8

Какое из утверждений о первичном ключе НЕ верно?

1. Первичный ключ может содержать NULL значения.

   Ответ2) Первичный ключ НЕ может содержать NULL значений.

3.Первичный ключ содержит только уникальные значения.

4.Каждая таблица имеет первичный ключ.

Вопрос №9

Для чего применяются индексы?

1. Для выборки из нескольких таблиц в одном запросе.
2. Для уменьшения места, занимаемого таблицей.
3. Для восстановления после случайного изменения.

   Ответ 4)Для ускорения операций выборки.

Вопрос №10

Выберите верное утверждение относительно индекса:

Ответ: 1) Индекс позволяет ускорить выборку с тем полем, для которого он сделан.

2.Индекс ускоряет абсолютно любые запросы с таблицей.

3.Индекс позволяет сэкономить место, занимаемое таблицей.
4.Индекс ускоряет добавление записей в таблицу.

Вопрос №11
Что делает команда CREATE?
1. Такой команды не существует.
2. Может и создавать таблицу, и добавлять запись.
3. Добавляет запись.

Ответ: 4) Создаёт таблицу.

Вопрос №12
Как правильно вставлять запись в таблицу?
1. INSERT INTO my_table(id = 1, name= FirstName)
2. INSERT INTO my_table (id = 1, name= 'FirstName')
3. INSERT INTO my_table (id, name) VALUES (1, FirstName)

Ответ 4) INSERT INTO my_table (id, name) VALUES (1, 'FirstName')

Вопрос №13
Каким запросом можно удалить все записи из таблицы "my_table" (но не саму таблицу)?
1. DROP TABLE "my_table"
2. DELETE "my_table"

Ответ 3) DELETE FROM "my_table"
4.DELETE TABLE "my_table"

Задачи

Задача №1
Есть 2 таблицы:
a) *Students*

id	first_name
1	Сергей
2	Андрей
3	Николай
4	Михаил
5	Валера

b) *Students_details*

first_name	subject
Николай	Science
Игорь	NULL
Михаил	NULL
Сергей	Science

a) Напишите запрос, используя подзапрос, который выберет id и f*irst_name всех студентов для которых тема = Science*
SELECT * FROM students WHERE `first_name` IN (SELECT `first_name` FROM students_details WHERE students_details.subject='Science')
b) Напишите требуемый запрос в п.а, используя JOIN
SELECT students.*, students_details.* FROM students inner join students_details on students.first_name=students_details.first_name AND students_details.subject='Science'
c) Напишите требуемый запрос в п.а, не используя JOIN и подзапросы.
SELECT * FROM `students_details`, students where students_details.first_name = students.first_name and students_details.subject='Science'

Задача №2

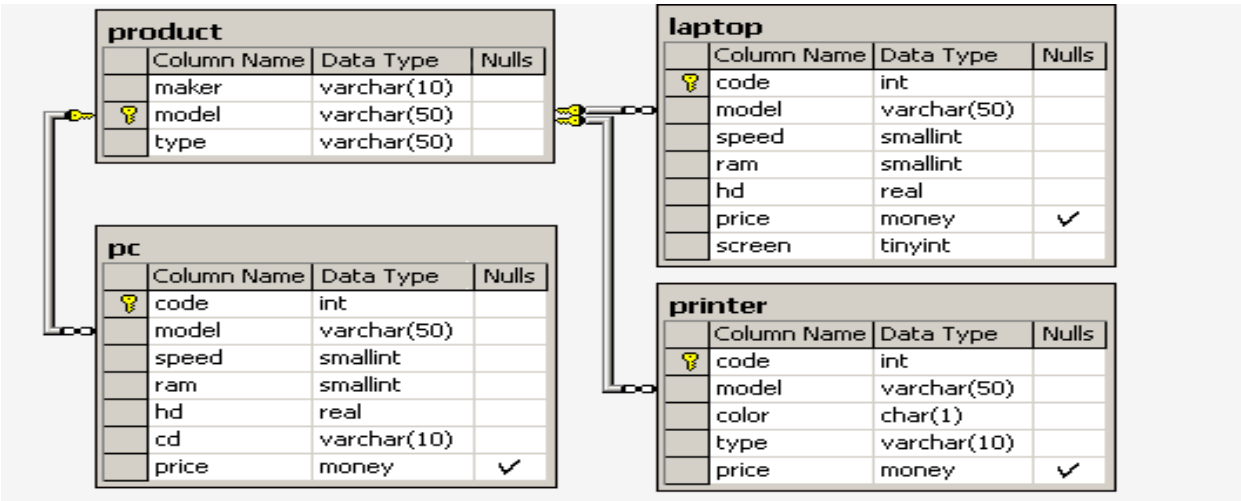
STREETS

id	Улица	Номер_Дома	Этажность
1	Платонова	1	12
2	Первая	2	2
3	Красная	3	6
4	Зеленая	4	45
5	Платонова	5	16
6	Зеленая	12	12

Напишите запрос

1. который выберет все улицы и дома, этажность которых больше 3 и меньше 16
   SELECT `Улица`,`Номер_Дома` FROM `streets` WHERE `Этажность` BETWEEN 3 AND 16
2. Который выберет максимальную этажность для каждой улицы, саму улицу и количество домов на каждой улице
   SELECT `Улица`,`Номер_Дома`, MAX(`Этажность`) as `Количество этажей` FROM `streets` GROUP BY `Улица`

Задача №3



1. Найдите номер модели, скорость , емкость жесткого диска (hd) для всех PCs, где цена ниже $500.
   SELECT model, speed, hd FROM pc WHERE price < 500
2. Найдите все цветные принтеры.
   SELECT * FROM Printer WHERE color != 'white'
3. Найдите номер модели и скорость всех PC которые имеют cd = 12x и их цена меньше чем $600, или cd =24x и цена меньше $700.
   SELECT model,speed,hd FROM pc WHERE (cd = '12x' OR cd = '24x') AND price < 700
4. Найдите производителя и скорость laptop, для которых емкость жесткого диска больше или равно 10 GB.
   SELECT DISTINCT pr.maker, lap.speed FROM laptop lap JOIN product pr ON pr.model = lap.model WHERE lap.hd >= 10
5. Найдите все модели и цены ВСЕХ продуктов , сделанных производителем B.
   SELECT DISTINCT product.model,pc.price FROM Product JOIN pc ON product.model = pc.model WHERE maker='B'
   UNION
   SELECT DISTINCT product.model,printer.price FROM product JOIN printer ON product.model=printer.model WHERE maker='B'
   UNION

SELECT DISTINCT product.model,laptop.price FROM product JOIN laptop ON product.model=laptop.model WHERE maker='B'

6. Найдите производителей, которые продают PCs но не продают laptops.
   SELECT DISTINCT maker FROM product WHERE type = 'pc'
   EXCEPT
   SELECT DISTINCT product.maker FROM product WHERE type = 'laptop'
7. Найдите принтеры с максимальной ценой
   SELECT model, price FROM printer WHERE price = (SELECT MAX(price)FROM printer)
8. Найдите среднюю скорость всех PCs, где цена больше 1000$
   SELECT AVG(speed) FROM pc WHERE price > 1000
9. Найдите модели laptops, которые имеют скорость меньше чем все PCs
   select prod.type,prod.model,lap.speed from laptop lap
   join product prod on lap.model=prod.model
   where lap.speed<(select min(speed) from pc)
10. Найдите производителей , которые производят хотя бы 3 модели PCs
    SELECT maker, COUNT(model) as kol_model FROM product WHERE type = 'pc' GROUP BY maker HAVING COUNT(model) >= 3
11. Найдите максимальную цену PCs для каждого производителя
    SELECT model FROM (SELECT model, price FROM pc UNION SELECT model, price FROM Laptop UNION SELECT model, price FROM Printer) a1
    WHERE price = (SELECT MAX(price) FROM (SELECT price FROM pc UNION SELECT price FROM Laptop UNION SELECT price FROM Printer) a2)

Задача №4

pc

	Column Name	Data Type	Nulls
🔑	code	int	
	model	varchar(50)	
	speed	smallint	
	ram	smallint	
	hd	real	
	cd	varchar(10)	
	price	money	✓

Напишите запрос, который вставит запись в эту таблицу.

```
INSERT INTO `pc`(`code`, `model`, `speed`, `ram`, `hd`, `cd`, `price`) VALUES ('1','i-5','1.6','8','500','ASUS','592,2')
```

Задача №5

printer

	Column Name	Data Type	Nulls
🔑	code	int	
	model	varchar(50)	
	color	char(1)	
	type	varchar(10)	
	price	money	✓

Удалите все принтеры из таблицы, у которых модель = «KLNM001» и цена меньше 150

```
DELETE FROM `printer` WHERE `model`= 'KLNM001' AND `price`<150
```