

CREATE SCHEMA `computer\_shop`;

DROP SCHEMA `computer\_shop`;

USE computer\_shop;

/\*

drop table Laptop;

drop table pc;

drop table printer;

drop table product;

\*/

-- USE computers;

CREATE TABLE Laptop (

code int NOT NULL ,

model varchar (50) NOT NULL ,

speed smallint NOT NULL ,

ram smallint NOT NULL ,

hd real NOT NULL ,

price decimal(12,2) NULL ,

screen tinyint NOT NULL

);

CREATE TABLE PC (

code int NOT NULL ,

model varchar (50) NOT NULL ,

speed smallint NOT NULL ,

ram smallint NOT NULL ,

hd real NOT NULL ,

cd varchar (10) NOT NULL ,

price decimal(12,2) NULL

);

CREATE TABLE Product (

maker varchar (10) NOT NULL ,

model varchar (50) NOT NULL ,

type varchar (50) NOT NULL

);

CREATE TABLE Printer (

code int NOT NULL ,

model varchar (50) NOT NULL ,

color char (1) NOT NULL ,

`type` varchar (10) NOT NULL ,

price decimal(12,2) NULL

);

ALTER TABLE Laptop ADD

CONSTRAINT PK\_Laptop PRIMARY KEY

(

code

);

ALTER TABLE PC ADD

CONSTRAINT PK\_pc PRIMARY KEY

(

code

);

ALTER TABLE Product ADD

CONSTRAINT PK\_product PRIMARY KEY

(

model

);

ALTER TABLE Printer ADD

CONSTRAINT PK\_printer PRIMARY KEY

(

code

);

ALTER TABLE Laptop ADD

CONSTRAINT FK\_Laptop\_product FOREIGN KEY

(

model

) REFERENCES Product (

model

);

ALTER TABLE PC ADD

CONSTRAINT FK\_pc\_product FOREIGN KEY

(

model

) REFERENCES Product (

model

);

ALTER TABLE Printer ADD

CONSTRAINT FK\_printer\_product FOREIGN KEY

(

model

) REFERENCES Product (

model

);

/\*----Product--------------------\*/

insert into Product values('B','1121','PC')

,('A','1232','PC')

,('A','1233','PC')

,('E','1260','PC')

,('A','1276','Printer')

,('D','1288','Printer')

,('A','1298','Laptop')

,('C','1321','Laptop')

,('A','1401','Printer')

,('A','1408','Printer')

,('D','1433','Printer')

,('E','1434','Printer')

,('B','1750','Laptop')

,('A','1752','Laptop')

,('E','2113','PC')

,('E','2112','PC');

/\*----PC-------------------------\*/

insert into PC values(1,'1232',500,64,5,'12x',600)

,(2,'1121',750,128,14,'40x',850)

,(3,'1233',500,64,5,'12x',600)

,(4,'1121',600,128,14,'40x',850)

,(5,'1121',600,128,8,'40x',850)

,(6,'1233',750,128,20,'50x',950)

,(7,'1232',500,32,10,'12x',400)

,(8,'1232',450,64,8,'24x',350)

,(9,'1232',450,32,10,'24x',350)

,(10,'1260',500,32,10,'12x',350)

,(11,'1233',900,128,40,'40x',980)

,(12,'1233',800,128,20,'50x',970)

,(13,'1121',900,64,5,'12x', 850)

,(14,'1121',900,127,11,'50x',850)

,(15,'1121',900,124,12,'40x',850)

,(16,'1121',900,138,14,'40x',850)

;

/\*----Laptop---------------------\*/

insert into Laptop values(1,'1298',350,32,4,700,11)

,(2,'1321',500,64,8,970,12)

,(3,'1750',750,128,12,1200,14)

,(4,'1298',600,64,10,1050,15)

,(5,'1752',750,128,10,1150,14)

,(6,'1298',450,64,10,950,12)

;

/\*----Printer--------------------\*/

insert into Printer values(1,'1276','n','Laser',400)

,(2,'1433','y','Jet',270)

,(3,'1434','y','Jet',290)

,(4,'1401','n','Matrix',150)

,(5,'1408','n','Matrix',270)

,(6,'1288','n','Laser',400)

;

/\* === === === === === === \*/

/\* > Агрегирующие функции - это функцие, которые

1. фильтруют значения

2. сортируют значения

3. работают с набором данных, превращая их в одно итоговое значение

4. суммируют все значения

\*/

/\* > Для подсчета количества записей в таблице

«Product» используется команда:

USE computer\_shop;

1. COUNT ROW IN Product;

2. SELECT COUNT(\*) FROM Product;

3. SELECT ROWS FROM Product;

4. SELECT SUM(\*) FROM Product;

\*/

/\* > Какая агрегатная функция используется для расчета суммы?

1. SUM

2. AVG

3. COUNT

\*/

/\* > Запрос для выборки первых 4 записей

из таблицы «Product» имеет вид

USE computer\_shop;

1. SELECT \* FROM Product LIMIT 4;

2. SELECT \* LIMIT 4 FROM Product;

3. SELECT \* FROM Product;

\*/

/\* > Что покажет SELECT-запрос?

SELECT

DISTINCT type

ORDER BY type

FROM Product;

1. Уникальные типы продуктов, отсортированные по возрастанию

2. Уникальные типы продуктов, отсортированные по убыванию

3. Ничего, запрос составлен неверно, ORDER BY всегда ставится в конце запроса

4. Неотсортированные никак уникальные типы продуктов

\*/

/\* > Что покажет следующий запрос:

SELECT \* FROM Product;

SELECT

maker,

count(\*) as type\_amount

FROM Product

GROUP BY maker

HAVING maker IN ("A", "C", "D")

ORDER BY type\_amount DESC --ASC

1. количество товаров сгруппированное по производителю A, C и D

2. количество товаров, у которых A, C и D производители

3. ничего, запрос составлен неверно, HAVING указывается до группировки

4. ничего, запрос составлен неверно, не хватает WHERE

5. Ничего, не хватает ; в конце

\*/

/\* Итог

SELECT

FROM

[WHERE]

[GROUP BY]

[HAVING]

[ORDER BY [ASC | DESC]]

\*/

/\*----View--------------------\*/

SELECT \* FROM Product;

SELECT \* FROM PC;

SELECT \* FROM Laptop;

SELECT \* FROM Printer;

/\*

Работа с таблицей PC

Добавить дополнительные записи

с повторяющимися speed и|или ram

\*/

/\* Тема «SELECT»

SELECT поле1, поле2, ...

FROM имя\_таблицы

WHERE условие

ORDER BY поле1, поле2, ... ASC(DESC);

1. Выведите все записи, отсортированные по полю "speed" по возрастанию

2. Выведите все записи, отсортированные по полю "model"

3. Выведите записи полей "code", "hd","price",

отсортированные по полю "price" в алфавитном порядке по убыванию

4. Выполните сортировку по полям "speed" и "ram" по убыванию

\*/

/\* Тема «Операторы сортировки. DISTINCT, LIMIT»

DISTINCT:

SELECT DISTINCT поле1, поле2, ...

FROM имя\_таблицы;

LIMIT:

LIMIT [смещение\_относительно\_начала, | количество\_извлекаемых\_строк]

1. Выведите уникальные (неповторяющиеся) значения поля «model»

2. Выведите первые две первые записи из таблицы

3. Пропустите первые 4 строки 1<=code<=4

и извлеките следующие 3 строки 5<=code<=7

✣ 4. Пропустите две последнии строки

и извлекаются следующие за ними 3 строки

Например, в таблице 10 строк 1<=code<=10.

Нужно пропустить 9<=code<=10. Показать 8>=code>=6

\*/

/\* Тема Агрегатные функции

SELECT \* FROM PC;

1. Рассчитайте общий бюджет. Каждая позиция встречается один раз

2. Выведите общую стоимость по моделям

3. Посчитайте количество записей в таблице

4. Выведите количество уникальных моделей

5. Найдите среднее стоимость по всем моделям

\*/

/\* Тема группировка

SELECT \* FROM PC;

1. Сгруппируйте поля по модели

Для каждой группы найдите суммарную стоимостью

2. Сгруппируйте поля по объёму ram

Найдите максимальный ценник внутри группы

3. Покажите Только модель с минимальным прайсом

\*/