SQL Заявки. Селектиране на данни

ЗАДАЧА:

Попълнете таблиците от базата данни от предходното упражнение по следния начин:

Табл.1. PRODUCTS

IDProduct	ProductName	Price
11 31 1 741 777	Deflection Rail	94.47$
11 31 1 745 406	Guild rail	31.48$
11 31 1 741 746	Timing chain	94.47$

Табл.2. CUSTOMERS

IDCustomers	Name	Email	Phone	Address
1	Denis	den@example.cm	111-22-33	n/a
2	Valeria	val@example.com	222-11-44	n/a
3	Max	max@example.com	555-11-11	New-York
4	Rusv	rusv@example.com	222-11-11	Sofia

Табл.3. ORDERS

IDOrders	IDCustomers	total	Payments	shipment
1	1	220.42$	credit_card	courier
12	2	188.94$	credit_card	SAT
13	3	94.47$	cash	SAT
14	3	188.94$	cash	SAT

Табл.4. DETAILS

IDOrders	IDProduct	IDCustomers
1	11 31 1 741 746	1
1	11 31 1 745 406	1
1	11 31 1 741 777	1
12	11 31 1 741 746	2
12	11 31 1 741 777	2
13	11 31 1 741 746	3
14	11 31 1 741 777	4
14	11 31 1 741 746	4

1. Оператор SELECT

Операторът SELECT избира записи, които съответстват на определени условия. Синтаксисът му е следния:

SELECT [DISTINCT] *[поле1 AS псевдоним, …. , полеN AS псевдонимN]*
FROM *Име_на_таблица1 [, … , Име_на_таблицаN]*
*[**WHERE** условие]*
*[**GROUP BY** списък с полета]*
*[**ORDER BY** списък с полета]*
*[**LIMIT** количество]*

Параметърът WHERE се използва за това какви полета трябва да бъдат селектирани. GROUP BY се използва за групиране на редовете по резултатите на следните функции (MAX, SUM,AVG и др.). ORDER BY се използва за подреждане на изходните данни. LIMIT се използва за уточняване колко на брой записа да се покажат. Задължителните параметри са SELECT и FROM.

За да тестваме възможностите на оператора SELECT нека да тестваме няколко примера:

SELECT *Name* ***FROM*** *CUSTOMERS* – селектира само имената на купувачите

SELECT *Name, Phone, Email, Address* ***FROM*** *CUSTOMERS* - селектира полетата в посочения ред

SELECT ******* ***FROM*** *CUSTOMERS – селектира всички полета от таблицата*

2. Задаване на условие в WHERE. Примери за тестване:

SELECT ***
FROM *CUSTOMERS*
WHERE *Address = ‘n/a’*

SELECT ***
FROM *CUSTOMERS*
WHERE *Name = ‘Max’*

3. Подреждане на резултатите:

```
SELECT *
FROM CUSTOMERS
ORDER BY Email DESC
```

```
SELECT *
FROM CUSTOMERS
ORDER BY Name  (ASC)
```

!Забележка: По подразбиране подреждането се осъществява по възходящ ред (ASC), но вие може да зададете и подреждане по низходящ ред (DESC).

4. Ограничаване на резултатите:

```
SELECT *
FROM CUSTOMERS
ORDER BY IDCustomers
LIMIT 2;
```

```
SELECT *
FROM CUSTOMERS
ORDER BY IDCustomers DESC
LIMIT 2;
```

5. Селектиране на случайни записи

```
SELECT *
FROM CUSTOMERS
ORDER BY RAND()
LIMIT 1;
```

Тествайте тази заявка няколко пъти и анализирайте резултата.

6. Специални функции:

```
SELECT SUM(total), max(total), min(total), avg(total)
FROM ORDERS
```

```
SELECT SUM(IDCustomers) as suma, address
FROM Customers
GROUP BY address
```

Помислете как да направим заявка която да изкарва по отделно общата сума платена чрез кредитна карта, и чрез кеш.

7. Селекция от повече от една таблица.

SELECT *Name, Total*
FROM *CUSTOMERS, ORDERS*
WHERE *CUSTOMERS.IDCustomers=Orders.IDCustomers*

8. Релационни оператори:

- = – равно;
- > – по-голямо от;
- < – по-малко от;
- >= – по-голямо или равно;
- <= – по-малко или равно;
- <> – не е равно;

SELECT ***
FROM *ORDERS*
WHERE *total>100;*

SELECT ***
FROM *ORDERS*
WHERE *total>=100*

9. Булеви оператори:

- AND – приема два булеви израза (A AND B) и връща TRUE ако и двата са истина;
- OR – Приема два булеви израза (A OR B) и връща TRUE ако единият е истина;
- NOT – Приема един израз (NOT A) в качеството на аргумент и променя неговата стойност в противоположната му (TRUE на FALSE, а FALSE на TRUE).

Примери:

SELECT ***
FROM *ORDERS*
WHERE *shipment = 'SAT'* ***AND*** *total>100;*

***SELECT** **
***FROM** ORDERS*
***WHERE** IDCustomers= 1 **AND** total>50;*

***SELECT** **
***FROM** ORDERS*
***WHERE NOT** payment = 'credit_card' **AND** total>100;*

***SELECT** **
***FROM** ORDERS*
***WHERE NOT** (payment = 'credit_card' **AND** total>100);*

Обърнете внимание на последните две заявки. Анализирайте ги.Изпълнете следващата заявка и я анализирайте с предходните две.

***SELECT** **
***FROM** ORDERS*
***WHERE NOT** payment = 'credit_card' **AND NOT** total>100;*

Анализирайте тази заявка със долната.

***SELECT** **
***FROM** ORDERS*
***WHERE NOT** payment = 'credit_card' **AND NOT** total<=100;*

ЗАДАЧИ:

Създайте базата данни KST_TEST. Създайте следните таблици към нея:
***Teachers** (TeachersID, First_Name, Last_Name, Phone, email),*
***Salary** (SalaryID, Salary),*
***Students** (StudentID, First_Name, Last_Name, Специаност, успех).*
Типа на полетата да отговаря спрямо тяхното предназначение. Попълнете с поне 4 или 5 записа всяка от таблиците. Изпълнете заявките които правят следните действия:

- *Селектирайте имената на учителите, емайл-ите и телефонните им номера в тази последователност. Пробвайте да зададете псевдоним по избор за някое от полетата.*
- *Селектирайте всички полета от таблица* Students.
- *Визуализирайте телефонните номера и имейлите на всички преподаватели които първото им име е „Иван“. Ако във вашата База Данни липсва такъв преподавател изберете такова име каквото имате в таблицата.*
- *Визуализирайте на всички студенти подредени по азбучен ред спрямо последното им име*
- *Визуализирайте всички данни на произволен студент;*
- *Изчислете сумата от заплатите в таблица* Salary.
- *Визуализирайте колко на брой студента има във всяка специалност.*
- *Визуализирайте всички студенти от специалност КСТ със успех 4 или по голям. (Посочете такава специалност и оценка в зависимост каквато има във вашата база данни.)*
- *Визуализирайте всички преподаватели с над 1200 лева и под 800 лева заплата. (Посочете такива суми че да имате преподаватели и в двата диапазона).*