**Shopping Management System (Music)**

**DESCRIPTION:**

In this Database Management System project, customers can buy Music albums online and the administrator can enter the name and generate the receipt of the purchased product. Administrator can view the customer ID, Order ID and date of purchase overall order details. Orders which are placed by the customers, will store into the database and according to the order detail, bill will be generated and the payment will be paid by the customer.

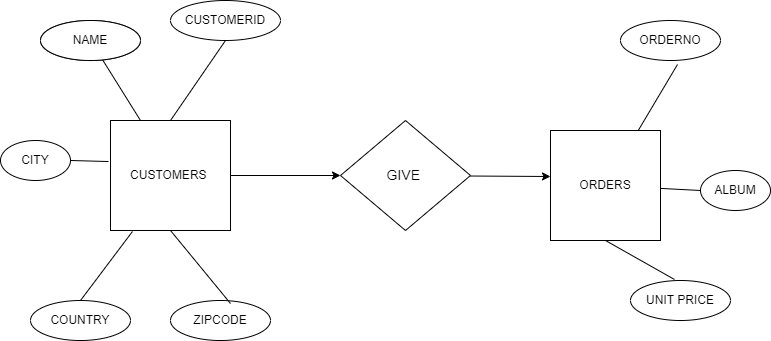
Some of the benefits of using this system are:

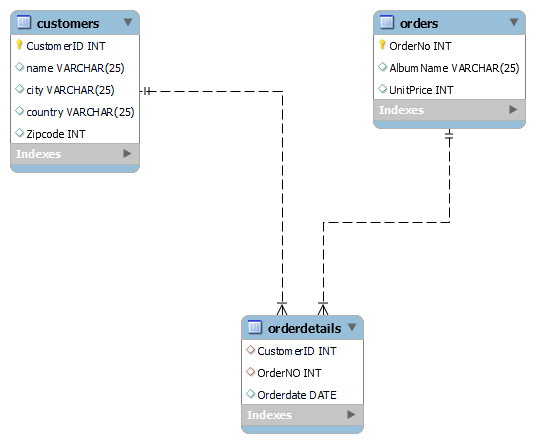
1. Update and modify data
2. Maintaining details accurate

This base contains total 3 tables:

1. CUSTOMERS
2. Orders
3. Order Details

**ER DIAGRAM:**





Commands:

Create database Furtado’s;

use Furtado’s;

create table customers

(

Customer ID int not null,

name varchar (25),

city varchar (25),

country varchar (25),

Zip code int,

primary key (Customer ID)

);

insert into customers (Customer ID, name, city, country, Zip code)

values

(1,"Zakaurkhan","Mumbai","India","400612"),

(2,"MarkTaylor","Sydney","Australia","2000"),

(3,"PhilHughes","London","Unitedkingdom","4567"),

(4,"SteveMurray","Edinburgh","UnitedKingdom","56789"),

(5,"Joakinjohnsson","Stockholm","SWEDEN","11127"),

(6,"HannahScheinder","Berlin","Germany","10789"),

(7,"AronMichelle","winnipeg","Canada","44455"),

(8,"CamilleBernard","Paris","France","70123");

select \*from customers;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CustomerID | name | city | country | Zipcode |
| 1 | Zakaurkhan | Mumbai | India | 400612 |
| 2 | MarkTaylor | Sydney | Australia | 2000 |
| 3 | PhilHughes | London | Unitedkingdom | 4567 |
| 4 | SteveMurray | Edinburgh | UnitedKingdom | 56789 |
| 5 | Joakinjohnsson | Stockholm | SWEDEN | 11127 |
| 6 | HannahScheinder | Berlin | Germany | 10789 |
| 7 | AronMichelle | winnipeg | Canada | 44455 |
| 8 | CamilleBernard | Paris | France | 70123 |

* Create table Orders

(

OrderNo int not null,

AlbumName Varchar(25),

UnitPrice int,

primary key(OrderNo)

);

Insert into Orders(OrderNo,AlbumName,UnitPrice)

values

(1001,"Presence",45000),

(2002,"HybridTheory",60000),

(3002,"Powerage",55000),

(4002,"Aerosmith",77000),

(5002,"LivingThings",95000),

(6002,"Madonna",70000);

select \*from Orders;

|  |  |  |
| --- | --- | --- |
| OrderNo | AlbumName | UnitPrice |
| 1001 | Presence | 45000 |
| 2002 | HybridTheory | 60000 |
| 3002 | Powerage | 55000 |
| 4002 | Aerosmith | 77000 |
| 5002 | LivingThings | 95000 |
| 6002 | Madonna | 70000 |

Create table OrderDetails

(

CustomerID int,

OrderNO int,

Orderdate date,

foreign key(CustomerID)references customers(CustomerID),

foreign key(OrderNO)references Orders(OrderNO)

);

insert into OrderDetails(CustomerID,OrderNO,Orderdate)

values

(1,1001,"2021-10-15"),

(2,2002,"2023-09-12"),

(3,2002,"2020-11-11"),

(4,3002,"2022-07-10"),

(5,4002,"2023-03-08");

select \*from OrderDetails;

|  |  |  |
| --- | --- | --- |
| CustomerID | OrderNO | Orderdate |
| 1 | 1001 | 15-10-2021 |
| 2 | 2002 | 12-09-2023 |
| 3 | 2002 | 11-11-2020 |
| 4 | 3002 | 10-07-2022 |
| 5 | 4002 | 08-03-2023 |

**MYSQL QUERIES:**

1. The following SQL statement selects all the customers from "CANADA":

Select \*from customers

**Where** state = “Canada”;

(Is used to filter records.)

OUTPUT:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7 | Aron Michelle | Winnipeg | Canada | 44455 |

1. The following SQL statement selects all fields from "Customers" where country is “UnitedKingdom" AND city is "Edinburgh":

select \*from customers

where country ="Unitedkingdom" and city ="Edinburgh";

(Operator displays a record if all the conditions separated by AND are TRUE)

Output:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | Steve Murray | Edinburgh | United Kingdom | 56789 |

1. The following SQL statement selects all customers from the "Customers" table, sorted DESCENDING by the "Country" column:

select \*from customers

order by country desc;

(Used to sort the result-set in ascending or descending order.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | Phil Hughes | London | United Kingdom | 4567 |
| 4 | Steve Murray | Edinburgh | United Kingdom | 56789 |
| 5 | Joakin Johansson | Stockholm | SWEDEN | 11127 |
| 1 | Zakaur khan | Mumbai | India | 400612 |
| 6 | Hannah Scheinder | Berlin | Germany | 10789 |
| 8 | Camille Bernard | Paris | France | 70123 |
| 7 | Aron Michelle | Winnipeg | Canada | 44455 |
| 2 | Mark Taylor | Sydney | Australia | 2000 |

1. The following SQL statement lists the number of customers in each country:

select count(CustomerID),country

from customers

group by country;

(Statement groups rows that have the same values into summary rows)

Output:

|  |  |
| --- | --- |
| 1 | Australia |
| 2 | Unitedkingdom |
| 1 | SWEDEN |
| 1 | Germany |
| 1 | Canada |
| 1 | France |

1. The following SQL statement lists the number of customers in each country. Only include countries with more than 2

customers:

SELECT COUNT(CUSTOMERID), COUNTRY

FROM CUSTOMER

GROUP BY COUNTRY

**HAVING** COUNT(CUSTOMERID)>2

ORDER BY COUNT(CUSTOMERID)>2 DESC;

Output:

|  |  |
| --- | --- |
| 2 | Unitedkingdom |

1. The following SQL statement will update the ZIP CODE to 400024 for all records where country is "UnitedKingdom":

update customers

set Zipcode = 7777

where country = "UnitedKingdom”;

(To modify the existing records in a table.)

OUTPUT:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Zakaurkhan | Mumbai | India | 400612 |
| 2 | MarkTaylor | Sydney | Australia | 2000 |
| 3 | PhilHughes | London | Unitedkingdom | 7777 |
| 4 | SteveMurray | Edinburgh | UnitedKingdom | 7777 |
| 5 | Joakinjohnsson | Stockholm | SWEDEN | 11127 |
| 6 | HannahScheinder | Berlin | Germany | 10789 |
| 7 | AronMichelle | winnipeg | Canada | 44455 |
| 8 | CamilleBernard | Paris | France | 70123 |

1. The following SQL statement selects all customers that are located in "Australia", "Unitedkingdom" or "Sweden":

SELECT \*FROM customers

where country in("Australia","Unitedkingdom","SWEDEN");

(Allows you to specify multiple values in a where clause.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | MarkTaylor | Sydney | Australia | 2000 |
| 3 | PhilHughes | London | Unitedkingdom | 7777 |
| 4 | SteveMurray | Edinburgh | UnitedKingdom | 7777 |
| 5 | Joakinjohnsson | Stockholm | SWEDEN | 11127 |

1. The following SQL statement selects all products with a price between 40000 to 100000

Select \*from orders

Where unit price **between** 40000 and 100000;

|  |  |  |
| --- | --- | --- |
| 1001 | Presence | 45000 |
| 2002 | HybridTheory | 60000 |
| 3002 | Powerage | 55000 |
| 4002 | Aerosmith | 77000 |
| 5002 | LivingThings | 95000 |
| 6002 | Madonna | 70000 |

(Operator selects values within a given range)

* **JOINS:**

**INNER JOIN :**

SELECT \*FROM customers inner join OrderDetails

on customers.CustomerID = OrderDetails.CustomerID;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CustomerID | name | city | country | Zipcode | CustomerID | OrderNO | Orderdate |
| 1 | Zakaurkhan | Mumbai | India | 400612 | 1 | 1001 | 15-10-2021 |
| 2 | MarkTaylor | Sydney | Australia | 2000 | 2 | 2002 | 12-09-2023 |
| 3 | PhilHughes | London | Unitedkingdom | 4567 | 3 | 2002 | 11-11-2020 |
| 4 | SteveMurray | Edinburgh | UnitedKingdom | 56789 | 4 | 3002 | 10-07-2022 |
| 5 | Joakinjohnsson | Stockholm | SWEDEN | 11127 | 5 | 4002 | 08-03-2023 |

**LEFT JOIN :**

SELECT \*FROM customers left join OrderDetails

on customers.CustomerID = OrderDetails.CustomerID;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CustomerID | name | city | country | Zipcode | CustomerID | OrderNO | Orderdate |
| 1 | Zakaurkhan | Mumbai | India | 400612 | 1 | 1001 | 15-10-2021 |
| 2 | MarkTaylor | Sydney | Australia | 2000 | 2 | 2002 | 12-09-2023 |
| 3 | PhilHughes | London | Unitedkingdom | 4567 | 3 | 2002 | 11-11-2020 |
| 4 | SteveMurray | Edinburgh | UnitedKingdom | 56789 | 4 | 3002 | 10-07-2022 |
| 5 | Joakinjohnsson | Stockholm | SWEDEN | 11127 | 5 | 4002 | 08-03-2023 |
| 6 | HannahScheinder | Berlin | Germany | 10789 | NULL | NULL | NULL |
| 7 | AronMichelle | winnipeg | Canada | 44455 | NULL | NULL | NULL |
| 8 | CamilleBernard | Paris | France | 70123 | NULL | NULL | NULL |

**RIGHT JOIN:**

SELECT \*FROM customers right join OrderDetails

on customers.CustomerID = OrderDetails.CustomerID;

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CustomerID | name | city | country | Zipcode | CustomerID | OrderNO | Orderdate |
| 1 | Zakaurkhan | Mumbai | India | 400612 | 1 | 1001 | 15-10-2021 |
| 2 | MarkTaylor | Sydney | Australia | 2000 | 2 | 2002 | 12-09-2023 |
| 3 | PhilHughes | London | Unitedkingdom | 4567 | 3 | 2002 | 11-11-2020 |
| 4 | SteveMurray | Edinburgh | UnitedKingdom | 56789 | 4 | 3002 | 10-07-2022 |
| 5 | Joakinjohnsson | Stockholm | SWEDEN | 11127 | 5 | 4002 | 08-03-2023 |

**Sub-queries:**

1. Find the name of customers from two tables without using joins

select name from customers

where CustomerID in (select CustomerID from OrderDetails where

OrderNo in (1001,2002));

Subquery: select CustomerID from OrderDetails where

OrderNo in(1001,2002)

|  |
| --- |
| Zakaurkhan |
| MarkTaylor |
| PhilHughes |

OutPut :

1. DISPLAY ALBUM NAME FOR CUSTOMER WHOS CUSTOMERID IS (1,2,3,4,5) WITHOUT USING JOINS

select AlbumName from Orders

where OrderNo in (select OrderNo from OrderDetails where CustomerID

= (1,2,3,4,5))

|  |
| --- |
| Presence |
| HybridTheory |
| Powerage |
| Aerosmith |

Output: