A Project Report on

**“Ice-cream Parlour Management System”**

**By**

|  |  |
| --- | --- |
| **Mr. Swapnil Valwe** | **200940581034** |
| **Ms. Meghna Patil** | **200940381049** |



**C-DAC Mumbai**

**ABSTRACT**

Management of Ice-Cream Parlour could be a tedious task on paper. To overcome this, there is advancement in Database Management System to create databases and manage it with ease.

In “Ice-Cream Parlour Management System” We have created an Entity-Relationship (ER) Diagram to map out the conceptual view of the Database. We have created 9 tables in a database. We have performed various Queries and have used Stored Procedures, Functions, Triggers, Cursors and Handlers to the best of our knowledge.

Hence this small project on Ice-Cream Parlour Management System would be useful for the better management.

**Keywords**-ER Diagram, Conceptual View,9 tables, Queries and have used Stored Procedures, Functions, Triggers, Cursors and Handlers.

**Entity Relationship (ER) Diagram:**

**Shapes Used in ER Diagram:**

Denotes Attributes of the Table.

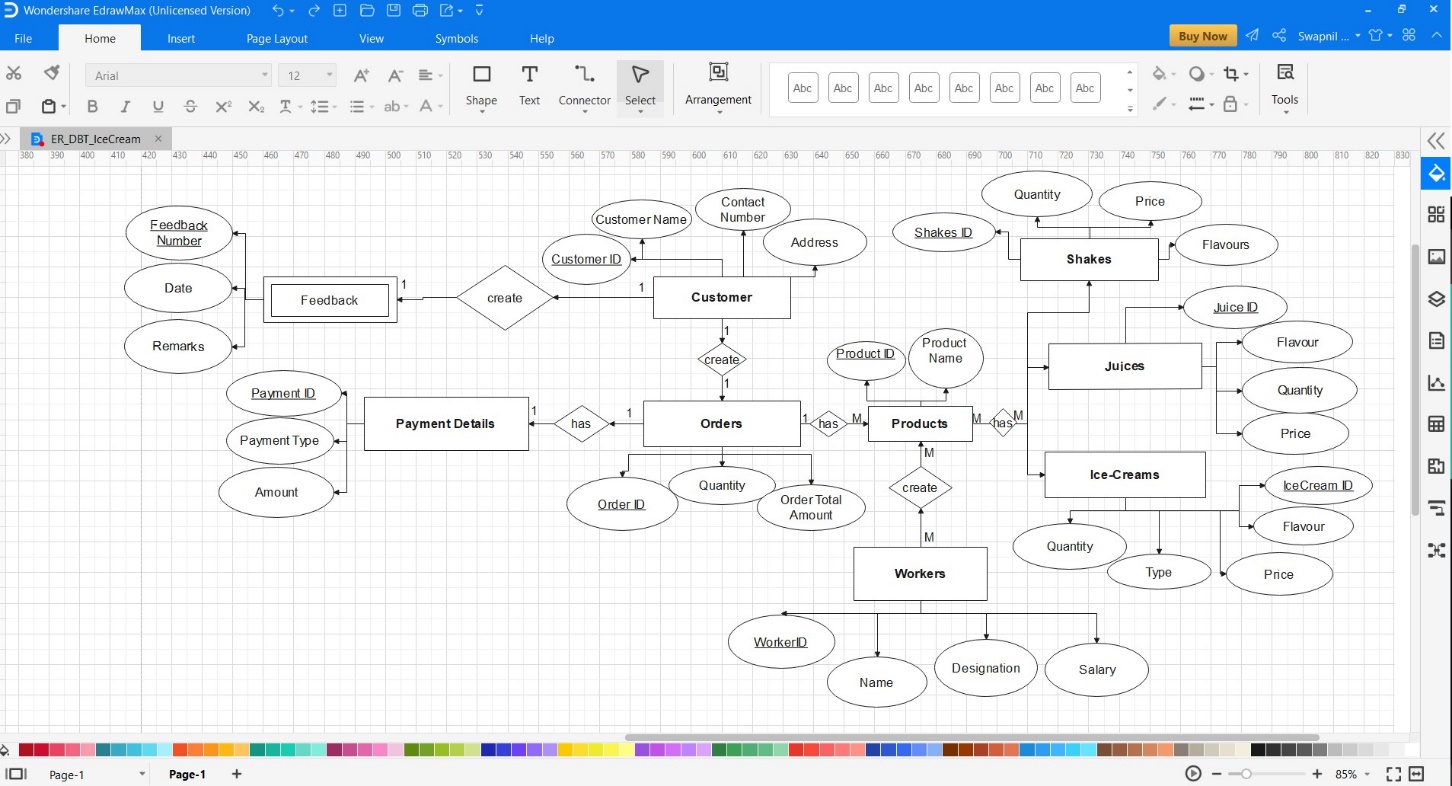
Denotes Entities of the Table.

Denotes Weak Entity of the Table.

Denotes Relationships between the two Entities.

Denotes Link between Entity, attributes Relations.

**ER Model For Ice-Cream Parlour Management System:**



**Number of Tables:**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Table Name** | **No. of the Attributes.** |
| 1. | Customer | 4 |
| 2. | Products | 2+3(Foreign keys) |
| 3. | Ice-Creams | 5 |
| 4. | Juices | 4 |
| 5. | Shakes | 4 |
| 6. | Orders | 3+1(Foreign Key) |
| 7. | Workers | 4 |
| 8. | Payment Details | 3+1(Foreign Key) |
| 9. | Feedback | 3+1(Foreign Key) |

**Schema and Queries:**

create database Ice\_Cream\_Parlour;

use Ice\_Cream\_Parlour;

1. **Customer:**

create table Customer\_Details(Customer\_ID int primary key unique not null, Customer\_Name varchar(50),Contact\_No numeric(11),Address varchar(50));

insert into customer\_details values(1201, 'MEGHNA PATIL', 6573528737, "SION"),

(1202, 'UDIT NEMADE', 539753795, "DADAR"),

(1203, 'SWAPNIL ANDHALE', 3856783267, "ANDHERI"),

(1204, 'TEJAS RANE', 9287389683, "DOMBIVLI"),

(1205, 'TANUJA PATIL', 1384536875, "BORIVALI"),

(1206, 'UNNATI TAPPE', 86738495273, "COLABA"),

(1207, 'MRUNALI DESHMUKH', 85398265483, "GHATKOPAR"),

(1208, 'MANOJ BHOYE', 36329759358, "THANE"),

(1209, 'UMESH CHAUDHARY', 7862387468, "MULUND"),

(1210, 'SWAPNIL VALWE', 8276863857, "BANDRA");

insert into customer\_details values (1212, 'KIRTI ROY', 9337568347, "MATUNGA");

|  |  |  |
| --- | --- | --- |
| **Customer** | | |
| Name | Data-Type | Description |
|  |  |  |
| Customer\_ID | Int | Primary Key |
| Customer\_Name | Varchar | Name of the Customers |
| Contact Number | numeric | Phone number |
| Address | varchar | Address of the customers |

1. **Orders:**

create table Orders(Order\_ID int primary key unique not null,Order\_Quantity numeric(5),Order\_total\_Amount numeric(4), Customer\_ID int, Product\_ID int);

alter table Orders add constraint Customer\_ID\_FK1 foreign key (Customer\_ID) references Customer\_Details(Customer\_ID);

alter table Orders add constraint Product\_ID\_FK1 foreign key (Product\_ID) references Products(Product\_ID);

insert into Orders values (9001, 4, 390, 1201, 1001),

(9002, 3, 280, 1204, 2001),

(9003, 2, 200, 1202, 2004),

(9004, 5, 430, 1206, 1005),

(9005, 7, 890, 1209, 3001);

select \* from Orders;

|  |  |  |
| --- | --- | --- |
| **Orders** | | |
| Name | Data-Type | Description |
|  |  |  |
| Order\_ID | Int | Primary Key |
| Order\_Quantity | Numeric | Quantity of the ordered Products |
| Order\_Total\_Amount | numeric | Total Price |
| Customer-ID | Int | Foreign Key |
| Product\_ID | Int | Foreign Key |

1. **Ice-Cream:**

create table Ice\_Cream(Ice\_Cream\_ID int primary key not null unique, Flavours varchar(50),Price numeric(4),Type varchar(50),Quantity numeric(5));

insert into Ice\_Cream values (101,'VANILLA',40,'CUP',70);

insert into Ice\_Cream values (102,'BUTTERSCOTCH',180,'FAMILYPACK',30);

insert into Ice\_Cream values (103,'ANJEER',200,'COCKTAILS',54);

insert into Ice\_Cream values (104,'REDVELVET',80,'CORNETTO',60);

insert into Ice\_Cream values (105,'MINT',50,'SCOOP',45);

|  |  |  |
| --- | --- | --- |
| **Ice-Cream** | | |
| Name | Data-Type | Description |
|  |  |  |
| Ice\_Cream\_ID | Int | Primary Key |
| Flavours | Varchar | Flavour of the Ice-Cream |
| Price | numeric | Amount |
| Type | Varchar | Type of Ice-Cream |
| Quantity | varchar | Quantity of the Ice-Cream |

1. **Juices:**

create table Juices(Juice\_ID int primary key unique not null,Flavour varchar(50),Price numeric(5),Quantity varchar(10));

insert into Juices values(200,'APPLE',120,'full'),

(201,'PINEAPPLE',50,'HALF'),

(202,'KIWI',150,'FULL'),

(203,'SWEETLIME',100,'FULL'),

(204,'ORANGE',45,'HALF'),

(205,'WATERMELON',110,'FULL');

|  |  |  |
| --- | --- | --- |
| **Juices** | | |
| Name | Data-Type | Description |
|  |  |  |
| Juice\_ID | Int | Primary Key |
| Flavour | Varchar | Flavour of the Juice |
| Price | numeric | Amount |
| Quantity | varchar | Half or Full |

1. **Shakes:**

create table Shakes(Shakes\_ID int primary key not null unique,Flavours varchar(50),Price numeric(5),Quantity varchar(10));

insert into Shakes values(300,'BADAM',90,'FULL'),

(301,'STRAWBERRY',70,'HALF'),

(302,'OREO',180,'FULL'),

(303,'WHITECHOCOLATE',150,'FULL'),

(304,'COLD-COFEE',80,'HALF');

|  |  |  |
| --- | --- | --- |
| **Shakes** | | |
| Name | Data-Type | Description |
|  |  |  |
| Shakes\_ID | Int | Primary Key |
| Flavour | Varchar | Flavour of the Shakes |
| Price | numeric | Amount |
| Quantity | varchar | Half or Full |

1. **Orders:**

create table Orders(Order\_ID int primary key unique not null,Order\_Quantity numeric(5),Order\_total\_Amount numeric(4), Customer\_ID int, Product\_ID int);

alter table Orders add constraint Customer\_ID\_FK1 foreign key (Customer\_ID) references Customer\_Details(Customer\_ID);

alter table Orders add constraint Product\_ID\_FK1 foreign key (Product\_ID) references Products(Product\_ID);

insert into Orders values (9001, 4, 390, 1201, 1001),

(9002, 3, 280, 1204, 2001),

(9003, 2, 200, 1202, 2004),

(9004, 5, 430, 1206, 1005),

(9005, 7, 890, 1209, 3001);

select \* from Orders;

|  |  |  |
| --- | --- | --- |
| **Orders** | | |
| Name | Data-Type | Description |
|  |  |  |
| Customer\_ID | Int | Primary Key |
| Customer\_Name | Varchar | Name of the Customers |
| Contact Number | numeric | Phone number |
| Address | varchar | Address of the customers |
| Product\_ID | Int | Foreign Key |
| Customer\_ID | Int | Foreign Key |

1. **Workers:**

create table Workers(Worker\_ID int primary key unique not null,Worker\_Name varchar(50),Worker\_Designation varchar(50),Worker\_Salary numeric(5));

insert into Workers values(20000,'SUHAS','CASHIER',18000),

(20001,'SHYAM','HELPER',1000),

(20002,'RAJ','BARISTA',14000),

(20003, 'SUPEKAR', 'MANAGER',25000);

|  |  |  |
| --- | --- | --- |
| **Workers** | | |
| Name | Data-Type | Description |
|  |  |  |
| Worker\_ID | Int | Primary Key |
| Worker\_Name | Varchar | Name of the Worker |
| Worker\_Designation | varchar | Post of the Worker |
| Worker\_Salary | numeric | Pay of the Worker |

1. **Payment Details:**

create table Payment\_Details(Payment\_ID int primary key unique not null, Payment\_Type varchar(40),Amount numeric(5), Order\_ID int);

alter table Payment\_Details add constraint Order\_ID\_FK foreign key (Order\_ID) references Orders(Order\_ID);

insert into Payment\_Details values (8011, "CC", 430, 9004),

(8012, "Gpay", 890, 9005),

(8013, "Cash", 200, 9003),

(8014, "Debit Card", 390, 9001),

(8015, "PhonePay", 280, 9002);

select \* from payment\_details;

|  |  |  |
| --- | --- | --- |
| **Payment Details** | | |
| Name | Data-Type | Description |
|  |  |  |
| Payment\_ID | Int | Primary Key |
| Payment\_Type | Varchar | Payment Type |
| Amount | numeric | Amount of the Order |
| Order\_ID | varchar | Foreign Key |

1. **Feedback:**

create table Feedback(FeedBack\_number int primary key unique not null, date date,Remark varchar(50), Customer\_ID int);

alter table Feedback add constraint Customer\_ID\_FK foreign key (Customer\_ID) references Customer\_Details(Customer\_ID);

insert into feedback values(1, '2019-06-26', "VERY GOOD", 1201),

(2, '2019-08-12', "GOOD", 1203),

(3, '2020-03-05', "VERY GOOD", 1206),

(4, '2020-07-21', "OKAY", 1210);

SELECT \* FROM FEEDBACK;

SELECT \* FROM CUSTOMER\_dETAILS INNER JOIN FEEDBACK ON customer\_details.CUSTOMER\_id = FEEDBACK.CUSTOMER\_ID;

|  |  |  |
| --- | --- | --- |
| **Feedback** | | |
| Name | Data-Type | Description |
|  |  |  |
| Feedback\_number | Int | Primary Key |
| Remark | Varchar | Remark from the Customer |
| date | date | Date of the feeback |
| Customer\_ID | int | Foreign Key |

**Stored Procedure for adding values into Customer\_details:**

delimiter //

create procedure Add\_Customer (

in Customer\_ID int (5),

in Customer\_name varchar(50),

in Contact\_no int (10),

in Place varchar(20)

)

begin

insert into customer\_Details values (customer\_ID, Customer\_name, contact\_no, place);

end //

call add\_Customer (1214, 'UDBHAV GARG', 375938759, 'DELHI');

**Using Functions for showing Customer\_Details**

delimiter &&

CREATE FUNCTION show\_CustomerDetails(

) returns int(50)

begin

declare abcd int(10);

set abcd =

(select count(Customer\_name) from Customer\_Details);

return abcd;

end &&

select show\_CustomerDetails();

**Using triggers for increasing counter**

set @counter = 13;

delimiter $$

create trigger countofCustomers after insert on Customer\_Details

for each row

begin

set @counter = @counter + 1;

end $$

select @counter;

**cursor 1 to show JuiceDetails using JuiceID**

DELIMITER &&

CREATE PROCEDURE SHOW\_JuicesDetails(IN ID int)

BEGIN

DECLARE FLAVOURS VARCHAR(20);

DECLARE PRICES INT(5);

DECLARE FLAG INT(2);

DECLARE SHOW\_CURSOR CURSOR FOR

SELECT FLAVOUR, PRICE FROM JUICES WHERE JUICE\_ID = ID;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET FLAG=1;

OPEN SHOW\_CURSOR;

SET FLAG=0;

FETCH SHOW\_CURSOR INTO FLAVOURS, PRICES;

WHILE(FLAG=0)

DO

FETCH SHOW\_CURSOR INTO FLAVOURS, PRICES;

SELECT FLAVOURS AS 'FLAVOUR',PRICES AS 'PRICE';

END WHILE;

CLOSE SHOW\_CURSOR;

END &&

DELIMITER ;

CALL SHOW\_JUICESDETAILS(200);

**cursor 2 to show JuiceDetails using Quantity**

DELIMITER &&

CREATE PROCEDURE SHOW\_QUANTITY(IN QUANTITY\_CURSOR varchar(20))

BEGIN

DECLARE FLAVOURS VARCHAR(20);

DECLARE PRICES INT(5);

DECLARE FLAG INT(2);

DECLARE SHOW\_CURSOR CURSOR FOR

SELECT FLAVOUR, PRICE FROM JUICES WHERE QUANTITY = QUANTITY\_CURSOR;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET FLAG=1;

OPEN SHOW\_CURSOR;

SET FLAG=0;

FETCH SHOW\_CURSOR INTO FLAVOURS, PRICES;

WHILE(FLAG=0)

DO

FETCH SHOW\_CURSOR INTO FLAVOURS, PRICES;

SELECT FLAVOURS AS 'FLAVOUR',PRICES AS 'PRICE';

END WHILE;

CLOSE SHOW\_CURSOR;

END &&

DELIMITER ;

CALL SHOW\_QUANTITY("FULL");