

Retail Mart Management

Problem statement:

A data analyst of a retail shop, Happy Mart, wants to store the product details, customer details, and order details to provide daily insights about customer behavior and product stock details.

Objective:

The objective is to design a database to easily evaluate and identify the performance of the shop to increase the daily sales.

1. -- create a database named SQL basics:

```
create database sqlbasics;
```

2. -- select SQL basics:

```
use sqlbasics;
```

3. -- create a product table:

```
create table product  
(p_code varchar(10) not null,  
p_name varchar(50),  
price int,  
stock varchar(50),  
category varchar(50),  
primary key (p_code));
```

4. -- create a customer table:

```
create table customer  
(c_id varchar(10) not null,  
c_name varchar(50),  
c_location varchar(50),  
c_phn int,  
primary key (c_id));
```

5. -- create a sales table:

```
create table sales  
(order_date date,  
order_number varchar(50) not null,  
p_code varchar(50),  
p_name varchar(50),  
quantity int,  
price int,  
primary key (order_number));
```

6. -- Insert values into customer table:

```
insert into customer (c_id, c_name, c_location, c_phn)
values ('1111','Nisha','Kerala',87456784),
('1212','Oliver','Kerala',73654890),
('1216','Nila','Delhi',87654398),
('1246','Vignesh','Chennai',65478904),
('1313','Shini','Maharashtra',84657389);
```

7. -- Insert values into sales table:

```
insert into sales (order_date, order_number,p_code, p_name,quantity,price)
values ('2021-02-10','HM04','25','conditioner',5,1000),
('2018-04-12','HM03','20','kiwi',3,420),
('2016-07-24','HM06','11','pencil',3,30),
('2019-01-11','HM07','19','apple',5,600),
('2016-10-19','HM09','17','biscuits',10,600);
```

8. -- Insert values into product table:

```
insert into product (p_code, p_name, price, stock, category)
values ('4','lays',10,20,'snacks'),
('11','pencil',4,10,'stationary'),
('17','biscuits',60,26,'snacks'),
('19','apple',120,9,'fruits'),
('20','kiwi',140,4,'fruits'),
('25','conditioner',200,5,'hair product'),
('26','oil bottle',40,2,'kitchen utencil');
```

9. -- Write a query to add new columns, such as serial number and categories, to the sales table:

```
alter table sales
add column (s_no int, categories varchar(20));
```

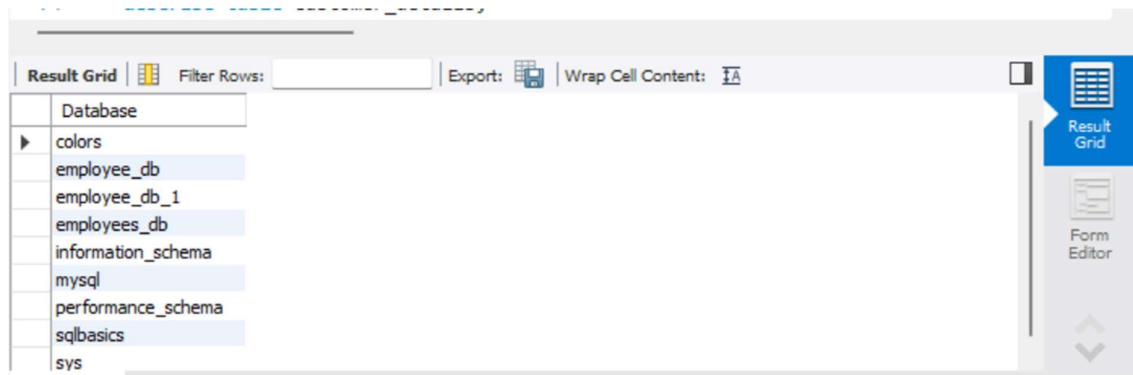
10. -- Write a query to change the stock field type to integer in the product table:

```
alter table product
modify stock int;
```

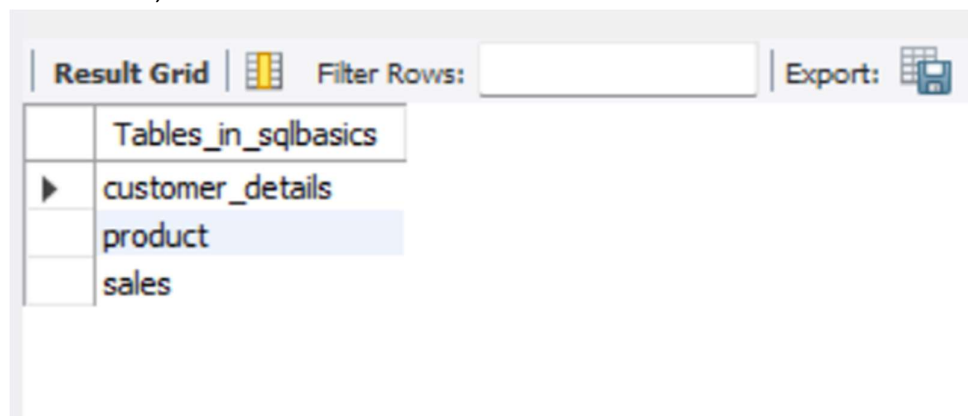
11. -- Write a query to change the table name from customer to customer details:

```
alter table customer
rename to customer_details;
```

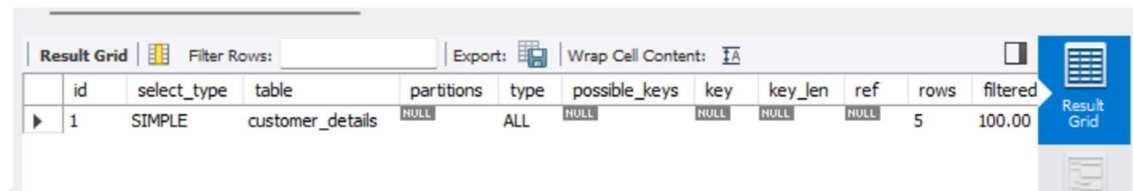
12. show databases;



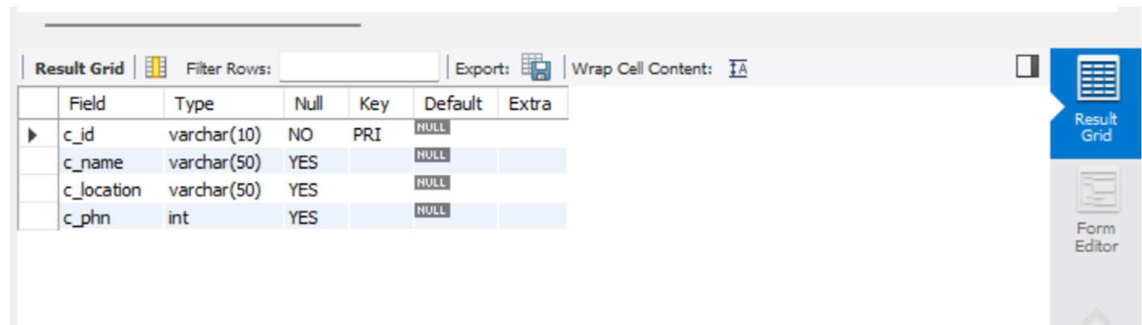
13. show tables;



14. describe table customer_details;



15. describe customer_details;



16. select*from customer_details;

c_id	c_name	c_location	c_phn
1111	Nisha	Kerala	87456784
1212	Oliver	Kerala	73654890
1216	Nila	Delhi	87654398
1246	Vignesh	Chennai	65478904
1313	Shini	Maharashtra	84657389
NULL	NULL	NULL	NULL

17. select*from product;

p_code	p_name	price	stock	category
11	pencil	4	10	stationary
17	biscuits	60	26	snacks
19	apple	120	9	fruits
20	kiwi	140	4	fruits
25	conditioner	200	5	hair product
26	oil bottle	40	2	kitchen utencil
4	lays	10	20	snacks
NULL	NULL	NULL	NULL	NULL

18. select*from sales;

order_date	order_number	p_code	p_name	quantity	price	s_no	categories
2018-04-12	HM03	20	kiwi	3	420	NULL	NULL
2021-02-10	HM04	25	conditioner	5	1000	NULL	NULL
2016-07-24	HM06	11	pencil	3	30	NULL	NULL
2019-01-11	HM07	19	apple	5	600	NULL	NULL
2016-10-19	HM09	17	biscuits	10	600	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

19. -- Write a query to drop the sl. no. and categories columns from the sales table:

```
alter table sales  
drop column s_no;  
alter table sales  
drop column categories;
```

20. -- Write a query to display the details where the category is stationary from the product table:

```
select * from product where category='stationary';
```

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
p_code	p_name	price	stock	category
11	pencil	4	10	stationary
NULL	NULL	NULL	NULL	NULL

21. -- Write a query to display the unique category from the product table:
 select distinct(category) from product;

Result Grid

22. -- Write a query to display the details of the sales from the sales table where
 quantity is greater than 2 and the price is less than 500:
 select* from sales where quantity>2 and price<500;

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

order_date

order_number

p_code

p_name

quantity

price

2018-04-12

HM03

20

kiwi

3

420

2016-07-24

HM06

11

pencil

3

30

*

NULL

NULL

NULL

NULL

NULL

NULL

Result Grid

Form Editor

23. -- Write a query to display every customer whose name ends with an 'a':
 select*from customer_details where c_name like '%a' ;

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

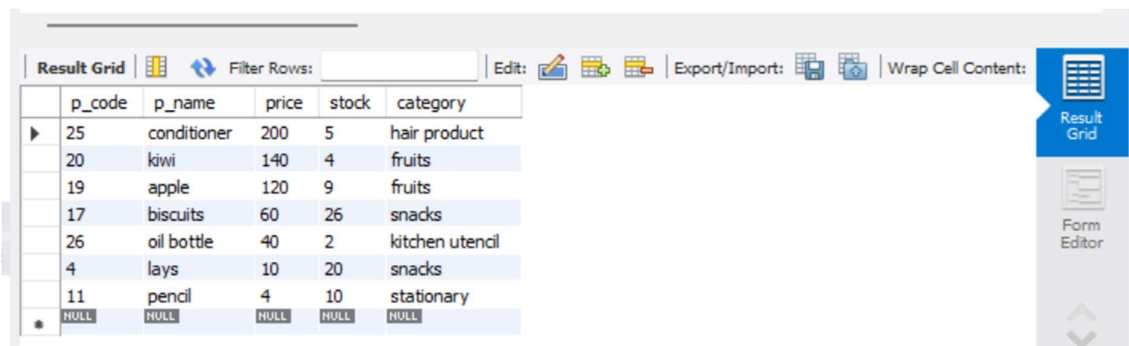
	c_id	c_name	c_location	c_phn
▶	1111	Nisha	Kerala	87456784
	1216	Nila	Delhi	87654398
*	NULL	NULL	NULL	NULL

Result Grid

Form Editor

24. -- Write a query to display the product details in descending order of price:

```
select * from product order by price desc;
```



	p_code	p_name	price	stock	category
▶	25	conditioner	200	5	hair product
	20	kiwi	140	4	fruits
	19	apple	120	9	fruits
	17	biscuits	60	26	snacks
	26	oil bottle	40	2	kitchen utensil
	4	lays	10	20	snacks
	11	pencil	4	10	stationary
*	NULL	NULL	NULL	NULL	NULL

25. -- Write a query to display the product code and category from categories that have two or more products:

```
select p_code, category from product group by category having count(category)>=2;
```

26. -- Write a query to combine the sales and product tables based on the order number and product's name:

```
select p.p_code, p.p_name, s.order_number from product p
left join sales s on p.p_code=s.p_code;
```



	p_code	p_name	order_number
▶	11	pencil	HM06
	17	biscuits	HM09
	19	apple	HM07
	20	kiwi	HM03
	25	conditioner	HM04
	26	oil bottle	NULL
	4	lays	NULL

27. -- Write a query to combine the sales and product tables based on the order number and customer's name including duplicated rows:

```
select p.p_code, p.p_name, s.order_number from product p
left join sales s on p.p_code=s.p_code
union all
select p.p_code, p.p_name, s.order_number from product p
right join sales s on p.p_code=s.p_code;
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	p_code	p_name	order_number
▶	11	pencil	HM06
	17	biscuits	HM09
	19	apple	HM07
	20	kiwi	HM03
	25	conditioner	HM04
	26	oil bottle	NULL
	4	lays	NULL
	20	kiwi	HM03
	25	conditioner	HM04
	11	pencil	HM06
	19	apple	HM07
	17	biscuits	HM09

Result Grid

Form Editor

Field Types

Query Stats

Result Grid

Form Editor

Field Types

Query Stats