**1. Write a SQL query to get the ProductID and Price for products with a Price of $100 or higher ?**

SELECT product\_id, price

FROM product

WHERE price >=100;

**2. Write a SQL query to get the customer and their corresponding online account?**

SELECT c.c\_name, o.username

FROM customer c, online\_cust n, online\_acc o

WHERE c.cust\_id = n.cust\_id

AND n.username = o.username;

**3. Write a SQL query to get the id, name and phone number of customers who have not placed orders ?**

**NO ROWS RETURNED**

SELECT c.cust\_id , c.c\_name, c.c\_phone

FROM customer c

LEFT JOIN cust\_invoice ON

c.cust\_id = cust\_invoice.cust\_id

WHERE cust\_invoice.cust\_id is NULL;

**4. Write a SQL query to list top 10 most expensive product?**

SELECT product\_id, price

FROM product

ORDER BY price DESC

LIMIT 10;

**5.** **List the products with product\_id beginning with the letter “C” and that cost at least $250 ?**

SELECT product\_id

FROM product

WHERE product\_id LIKE 'C%'

AND price > 249;

**6. Write a SQL query to list the stock type and product whose price is less than $500?**

SELECT s.stk\_type, p.product\_id

FROM stock s, product p, in\_stock i

WHERE s.stk\_id = i.stk\_id

AND i.product\_id = p.product\_id

AND price < 500;

**7. Write a SQL query to find the ID of each employee with no manager?**

select e\_id from employee

natural left outer join

works where manager\_id is null;

**8. Write a SQL query to find customer name who had purchaced any item ?**

SELECT c\_name

FROM customer LEFT JOIN cust\_invoice ON

cust\_invoice.cust\_id = customer.cust\_id

LEFT JOIN invoice ON

invoice.invoice\_id = cust\_Invoice.invoice\_id;

**9. Write a SQL query for each product that was sold in more than one sales transaction, retrieve the ProductID and the number of sales transactions in which the product was sold.**

**NO ROWS RETURNED:**

SELECT product.product\_id, Count(cust\_invoice.invoice\_id )AS Number\_of\_sales

FROM product, invoice, purchases, customer, cust\_invoice

WHERE product.product\_id = purchases.product\_id AND

purchases.cust\_id = customer.cust\_id AND

cust\_invoice.cust\_id = customer.cust\_id AND

invoice.invoice\_id = cust\_invoice.invoice\_id

GROUP BY product.product\_id

HAVING count(cust\_invoice.cust\_id) > 1;

**10. Write a SQL query to retrieve the product id, price of products**

**that belong to the store located in Garfield ?**

SELECT product.product\_id, product.price, store.s\_address

FROM product, store, contains

WHERE product.product\_id = contains.product\_id

AND store.store\_id = contains.store\_id

AND store.s\_address LIKE '%Garfield%';

**11.** **Write a SQL QUERY TO LIST ALL INVOICES ARE GREATER THAN $2500 INCLUDING CUSTOMER NAME AND CUSTOMER ID?**

SELECT customer.c\_name, customer.cust\_id,

invoice.total\_amount

FROM invoice INNER JOIN cust\_invoice ON

invoice.invoice\_id = cust\_invoice.invoice\_id

INNER JOIN customer ON

customer.cust\_id = cust\_invoice.cust\_id

GROUP BY customer.cust\_id

HAVING invoice.total\_amount>2500;

**12. Write a SQL query to find the products that are out of stock at every store ?**

FROM in\_stock, product, store, contains

WHERE in\_stock.product\_id = product.product\_id

AND contains.product\_id= product.product\_id

AND contains.store\_id = store.store\_id

Group by product.product\_id

Having product.total\_stock = 0;

**13. Write a SQL query showing the invoice id , the name of the customer that placed the invoice ?**

SELECT cust\_invoice.invoice\_id, customer.c\_name

FROM cust\_invoice

INNER JOIN invoice ON

invoice.invoice\_id = cust\_invoice.invoice\_id

INNER JOIN customer ON

customer.cust\_id = cust\_invoice.cust\_id;

**14.** **Retrieve the total number of customers ?**

SELECT count(\*) AS total\_customers

FROM customer;

**15. Write a SQL query that shows the IDs of employees in every store ?**

SELECT w.e\_id, w.store\_id

FROM employee e, works w, store s

WHERE e.e\_id = w.e\_id

AND w.store\_id = s.store\_id

ORDER BY s.store\_id DESC;

**16. Write a SQL query to find all the employees who are not managers?**

SELECT employee.e\_id, employee.e\_name

FROM employee LEFT JOIN

works ON employee.e\_id = works.e\_id

WHERE works.e\_id is NULL;

**17. Write a SQL query that shows productID for items that have an item cost between $750 and $1000 ?**

SELECT product\_id

FROM product

WHERE price >= 750 AND price <= 1000;

**18. Write a SQL query that shows e\_id and e\_name for employees whose names are Erica?**

SELECT e\_id, e\_name

FROM employee

where e\_name like 'Erica%';