DBMS PROJECT: SHOPPING MART - QUERIES

```sql
-- 1. Find the total sales revenue for each item:
SELECT i.item_code, i.item_name, SUM(bd.quantity * i.selling_price) AS total_sales_revenue
FROM item i
JOIN bill_details bd ON i.item_code = bd.item_code
GROUP BY i.item_code, i.item_name;

-- 2. Find the items with no active offers:
SELECT i.item_code, i.item_name
FROM item i
LEFT JOIN offers o ON i.item_code = o.item_code
WHERE o.item_code IS NULL;

-- 3. List the employees who have been with the company for more than 5 years:
SELECT emp_id, name, hire_date
FROM employee
WHERE hire_date <= CURRENT_DATE - INTERVAL '5 years';

-- 4. Find the items with a selling price greater than the average selling price of all items:
SELECT item_code, item_name, selling_price
FROM item
WHERE selling_price > (SELECT AVG(selling_price) FROM item);

-- 5. List the top 3 suppliers based on the total number of items they supply:
SELECT od.supplier_name, COUNT(*) AS total_items_supplied
FROM order_details od
GROUP BY od.supplier_name
ORDER BY total_items_supplied DESC
LIMIT 3;
```

```sql
-- 6. Find the employee with the longest tenure in the company:
SELECT emp_id, name, hire_date
FROM employee
ORDER BY hire_date ASC
LIMIT 1;
```

```sql
-- 7. Find the top 3 sections in terms of total revenue earned from item sales:
SELECT s.sid, s.sname, SUM(bd.quantity * i.selling_price) AS total_revenue
FROM section s
JOIN employee e ON s.mgr_id = e.emp_id
JOIN item i ON s.sid = i.sid
JOIN bill_details bd ON i.item_code = bd.item_code
GROUP BY s.sid, s.sname
ORDER BY total_revenue DESC
LIMIT 3;
```

```sql
-- 8. Identify the top 3 sections with the highest average employee salary:
SELECT s.sid, s.sname, AVG(e.salary) AS avg_salary
FROM section s
JOIN employee e ON s.mgr_id = e.emp_id
GROUP BY s.sid, s.sname
ORDER BY avg_salary DESC
LIMIT 3;
```

```sql
-- 9. Retrieve the names of employees whose birthdays fall within the next 30 days:
SELECT name, date_of_birth
FROM employee
WHERE EXTRACT(DAY FROM date_of_birth) BETWEEN EXTRACT(DAY FROM
CURRENT_DATE) AND EXTRACT(DAY FROM CURRENT_DATE + INTERVAL '30 days');
```

```
--10.List the top 3 suppliers based on the total number of items they supply:
SELECT s.supplier_name, COUNT(*) AS total_items_supplied
FROM supplier s
JOIN item i ON s.supplier_name = i.supplier_name
GROUP BY s.supplier_name
ORDER BY total_items_supplied DESC
LIMIT 3;
```

```
--11.Find the employee with the longest tenure in the company:
SELECT emp_id, name, hire_date
FROM employee
ORDER BY hire_date ASC
LIMIT 1;
```

```
--12.Find the top 3 most profitable sections in terms of total revenue earned from item sales:
SELECT s.sid, s.sname, SUM(bd.quantity * i.selling_price) AS total_revenue
FROM section s
JOIN item i ON s.sid = i.sid
JOIN bill_details bd ON i.item_code = bd.item_code
GROUP BY s.sid, s.sname
ORDER BY total_revenue DESC
LIMIT 3;
```

```
-- 13. Find the employees who have the highest salary in each shift:
SELECT shift, MAX(salary) AS highest_salary
FROM employee
GROUP BY shift;
```

```
--14.Calculate the total number of days each item has been in stock:
SELECT i.item_code, i.item_name,
CURRENT_DATE - od.order_date AS days_in_stock
FROM item i
JOIN order_details od ON i.item_code = od.item_code;

--15.Calculate the total number of days between the first and last purchase for each customer
SELECT Cust_id, cust_name,
EXTRACT(DAY FROM MAX(b.date) - MIN(b.date)) AS total_days_between_purchases
FROM customer c
JOIN bill b ON c.cust_id = b.cust_id
GROUP BY cust_id, cust_name;

--16.Identify the top 3 most common payment modes used by customers:
SELECT payment_mode, COUNT(*) AS num_customers
FROM customer
GROUP BY payment_mode
ORDER BY num_customers DESC
LIMIT 3;

--17. Retrieve the top 5 items with the highest quantity ordered
SELECT od.item_code, i.item_name, SUM(od.quantity_ordered) AS total_quantity_ordered
FROM order_details od
JOIN item i ON od.item_code = i.item_code
GROUP BY od.item_code, i.item_name
ORDER BY total_quantity_ordered DESC
LIMIT 5;

--18.Retrieve the top 5 items with the highest refund percentage in their return policy
SELECT rp.item_code, i.item_name, rp.refund_percent
FROM return_policy rp
JOIN item i ON rp.item_code = i.item_code
```

```
ORDER BY rp.refund_percent DESC
LIMIT 5

--19.Calculate the total warranty coverage period offered by each service provider
Page 3
SELECT service_provider, SUM(CAST(SPLIT_PART(warranty_period, ' ', 1) AS INTEGER)) AS
total_warranty_days
FROM warranty
GROUP BY service_provider
ORDER BY total_warranty_days DESC;

--20.Calculate the total number of items sold under each warranty period:
SELECT warranty_period, COUNT(*) AS total_items_sold
FROM warranty
JOIN item ON warranty.item_code = item.item_code
JOIN bill_details ON item.item_code = bill_details.item_code
GROUP BY warranty_period
ORDER BY total_items_sold DESC;

--21.Find the product that has arrvied last(newest product).
SELECT * FROM item i NATURAL JOIN order_details
ORDER BY arrival_date DESC
LIMIT 1;



--22.Identify the section with highest quantity sold in March 2024.
SELECT s.sid, s.sname, SUM(bd.quantity) AS quantity_sold
FROM bill
JOIN bill_details bd ON bill.bill_id = bd.bill_id
JOIN item ON bd.item_code = item.item_code
JOIN section s ON item.sid = s.sid
```

```sql
WHERE bill.date >= '2024-03-01' AND bill.date <= '2024-03-31'

GROUP BY s.sid, s.sname

ORDER BY quantity_sold DESC

LIMIT 1;


--23.List of dates with highest total sales(in terms of quantity) for March 2024.

select date, sum(quantity) as quantity_sold from (bill natural join bill_details)

where date>='2024-03-01' AND date<='2024-03-30'

group by date

order by quantity_sold DESC

limit 3;


--24.Retrieve the products that have been restocked in May, 2024.

select item_code ,item_name , quantity_ordered from order_details natural join item

where arrival_date>='2024-05-01' AND arrival_date<='2024-05-31';


--25.List the top 5 sections with the highest total revenue for the year 2024.

select item_code ,item_name ,quantity_ordered from order_details natural join item

where arrival_date>='2024-05-01' AND arrival_date<='2024-05-31';


--26.Find the supervisor whose salary is greater than all other supervisors.

select e2.emp_id as super_id,e2.salary as maximum_salary from employee as e1 join employee as e2 on e1.super_id=e2.emp_id

group by e2.emp_id

order by e2.salary DESC

limit 1;


--27.List of items that have been ordered but not yet delivered

SELECT * FROM order_details WHERE arrival_date IS NULL;

--List of orders that arrived within a month of being ordered

select * from order_details where (extract(month from arrival_date) - extract(month from order_date)=0)
```

```sql
AND (extract(year from arrival_date) - extract(year from order_date)=0) ;

--28.List of items that have been ordered more  than once in the year 2024.

SELECT item.item_code, item.item_name
FROM item
NATURAL JOIN order_details
WHERE extract(year from order_date) = 2024
GROUP BY item.item_code, item.item_name
HAVING count(distinct order_date) > 1;

--29.Top selling item (in terms of numbers) in  March 2024.
SELECT item_code,item_name,SUM(quantity) as total_quantity
FROM bill_details natural join item natural join bill
WHERE date >= '2024-03-01' AND date <= '2024-03-31'
GROUP BY item_code,item_name
ORDER BY total_quantity DESC
LIMIT 1;

--30.Find the customer who made the largest purchase(in terms of total purchase price).
SELECT cust_id,cust_name,SUM(selling_price*quantity) as total_cost
FROM bill natural join customer natural join bill_details natural join item
GROUP BY cust_id,cust_name
ORDER BY total_cost DESC
LIMIT 1;

--31.List employees with highest salary in each section.
SELECT emp_id,name,sid,sname,salary FROM employee as e natural join section
NATURAL JOIN (SELECT
sid, max(salary) as max_sal FROM employee GROUP BY sid) as av
WHERE salary = av.max_sal;
```

```
 --32.How many products belong to the 'Electronics' section?
SELECT COUNT(*) as no_of_product
FROM item i JOIN section s ON i.sid = s.sid
WHERE s.sname = 'electronics'; --5. List of employee who give more than one email_address in emp_email.
select emp_id,name,count(*) from employee natural join emp_email
group by emp_id,name
having count(*)>=2;
```

```
--33.Which section has the lowest total sales for a given date- ‘2024-03-07’?
SELECT sid, sname, SUM(selling_price*quantity) as total_sales
FROM bill natural join item natural join section natural join bill_details
WHERE date = '2024-03-07'
GROUP BY sid,sname
ORDER BY total_sales ASC
LIMIT 1;
```

```
--34.Find the customer who made the largest purchase (in terms of quantity).
SELECT cust_id,cust_name,SUM(quantity) as total_quantity
FROM bill natural join customer natural join bill_details
GROUP BY cust_id,cust_name
ORDER BY total_quantity DESC
LIMIT 1;
```

```
--35.Identify the products with the highest prices in each section.
select item_code,item_name,sid,sname from item as it natural join section natural join
(SELECT s.sid, s.sname, MAX(i.selling_price) as highest_price
FROM section s
JOIN item i ON s.sid = i.sid
GROUP BY s.sid, s.sname) as w where it.selling_price=highest_price;
```