Advanced Database Management Final Project

Courier management services

Group-8

(Accelerate)

Team Members:

Naveen Kumar Raghupathi - U33032085

Manish Boyina – U59955856

Raatan Kumar Garuda – U69871422

Varun Reddy Surakanti – U22657639

Raj Jeshwant Kumar Bandari – U08911814

Project Aim :

The Aim of this project is to develop a system that enhances a company's courier services.

Purpose:

This document shows the model courier management systems which we named Accelerate courier services that offer expedited shipping compared to standard shipping. The login facility for the admin will operate the system. While receiving the orders from the customers we'll collect the data such as customer name, customer contact number, and customer address. It generates the Tracking id, Where the customers can track their shipment from any location. It will provide the status of the consignment after placing the order within the stipulated time. The main section shows the price that is charged for each shipment to deliver which depends on the weight of the package.

Create Table:

```
create table login(
login_id varchar2(10) not null,
employee_id varchar2(20) not null,
password varchar2(20) not null,
PRIMARY KEY(login_id));

create table employee(
employee_id varchar2(20) not null,
login_id varchar2(10) not null,
employee_name varchar2(50) not null,
employee_phnumber varchar2(20) not null,
employee_email varchar2(50) not null,
PRIMARY KEY(employee_id));

create table item(
item_id varchar2(10) not null,
order_id varchar2(10) not null,
item_type varchar2(50) not null,
item_desc varchar2(20) not null,
item_weight varchar2(50) not null,
PRIMARY KEY(item_id));

create table delivery(
delivery_id varchar2(10) not null,
order_id varchar2(20) not null,
delivery_name varchar2(100) not null,
delivery_address1 varchar2(200) not null,
```

```
delivery_address2 varchar2(200) not null,
delivery_mobile varchar2(20) not null,
delivery_email varchar2(20) not null,
delivery_city varchar2(20) not null,
delivery_zipcode varchar2(20) not null,
PRIMARY KEY(delivery_id));

create table payment(
payment_id varchar2(10) not null,
order_id varchar2(10) not null,
payment_type varchar2(50) not null,
payment_status varchar2(20) not null,
delivery_price decimal(10,2) not null,
PRIMARY KEY(payment_id));

create table customer(
cust_id varchar2(10) not null,
order_id varchar2(10) not null,
cust_firstname varchar2(50) not null,
cust_lastname varchar2(20) not null,
cust_mobile varchar2(20) not null,
cust_address varchar2(50) not null,
cust_email varchar2(20) not null,
created_at timestamp not null,
PRIMARY KEY(cust_id));

create table tracking(
tracking_id varchar2(10) not null,
```

```
delivery_id varchar2(10) not null,
cust_id varchar2(50) not null,
order_id varchar2(20) not null,
tracking_status varchar2(20) not null,
PRIMARY KEY(tracking_id));

create table orders(
order_id varchar2(10) not null,
employee_id varchar2(10) not null,
item_id varchar2(50) not null,
payment_id varchar2(20) not null,
delivery_id varchar2(20) not null,
tracking_id varchar2(20) not null,
PRIMARY KEY(order_id));
```

Alter Table:

```
alter table employee add foreign key(login_id) references login(login_id);
alter table customer add foreign key(order_id) references orders(order_id);
alter table tracking add foreign key(delivery_id) references
delivery(delivery_id);
alter table tracking add foreign key(cust_id) references customer(cust_id);

alter table orders add foreign key(employee_id) references
employee(employee_id);
alter table orders add foreign key(item_id) references item(item_id);
alter table orders add foreign key(payment_id) references
payment(payment_id);
alter table orders add foreign key(delivery_id) references delivery(delivery_id);
```

alter table orders add foreign key(tracking_id) references tracking(tracking_id);

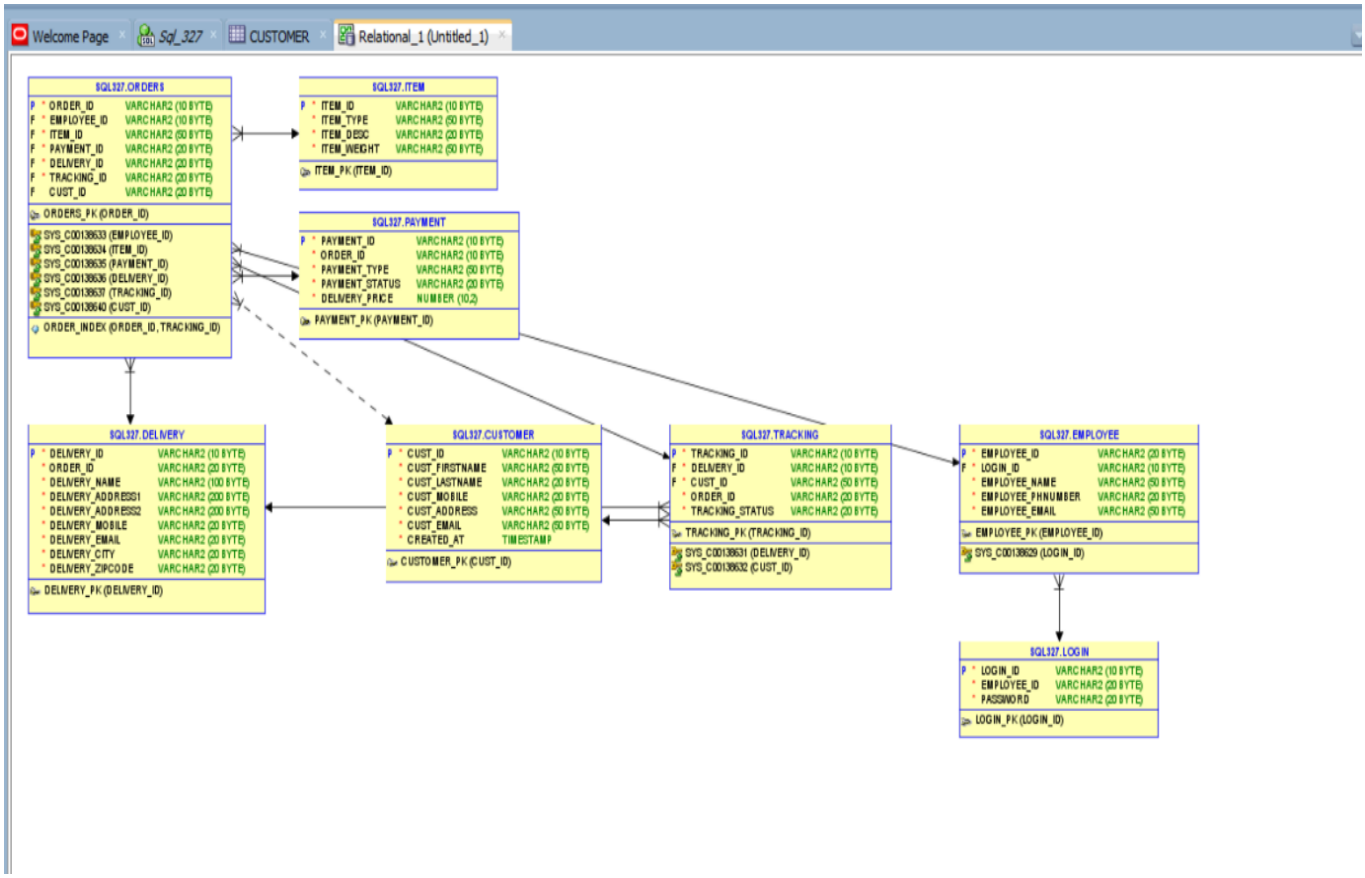
alter table customer modify cust_email varchar2(50);

alter table orders add cust_id varchar2(20);

alter table customer drop column order_id;

alter table orders add foreign key(cust_id) references customer(cust_id);

Entity Relationship Diagram Representing Database Design :



Query Writing:

Query 1:

We retrieved customer id, name and the related orders of the customer along with the employee details by using inner join on three tables that are - customer table, employee table and orders table.

```sql
SELECT customer.cust_id, customer.cust_firstname,
        orders.order_id, orders.item_id, orders.tracking_id, orders.delivery_id,
       employee.employee_id, employee.employee_name
FROM customer
INNER JOIN orders ON customer.cust_id = orders.cust_id
INNER JOIN employee ON employee.employee_id  = orders.employee_id;
```

Output:

Script Output | Explain Plan | Query Result

SQL | All Rows Fetched: 15 in 0.035 seconds

	CUST_ID	CUST_FIRSTNAME	ORDER_ID	ITEM_ID	TRACKING_ID	DELIVERY_ID	EMPLOYEE_ID	EMPLOYEE_NAME
1	cust013	vineeth	ord001	item001	track001	del001	30104630	Chris
2	cust005	sanath	ord002	item002	track002	del002	30104631	John
3	cust001	vamsi	ord003	item003	track003	del003	30104632	Angilis
4	cust006	kaundinya	ord004	item004	track004	del004	30104633	Lilian
5	cust007	sai	ord005	item005	track005	del005	30104634	Fedrica
6	cust004	suresh	ord006	item006	track006	del006	30104635	Juilian
7	cust008	sudeep	ord007	item007	track007	del007	30104636	Drake
8	cust003	karthik	ord008	item008	track008	del008	30104637	William
9	cust002	preetham	ord009	item009	track009	del009	30104638	Daniel
10	cust009	ram	ord010	item010	track010	del010	30104639	Danny
11	cust010	aditya	ord011	item011	track011	del011	30104640	Sammy
12	cust012	vidyut	ord012	item012	track012	del012	30104641	David
13	cust014	srinu	ord013	item013	track013	del013	30104642	Gayle
14	cust011	vidya	ord014	item014	track014	del014	30104643	Samantha
15	cust015	arshadeep	ord015	item015	track015	del015	30104644	Andrea

Script Output | Explain Plan | Query Result

SQL | 0.207 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			15	9
HASH JOIN			15	9
Access Predicates				
EMPLOYEE.EMPLOYEE_ID=ORDERS.EMPLOYEE_ID				
NESTED LOOPS			15	9
NESTED LOOPS				
STATISTICS COLLECTOR				
MERGE JOIN		COST=3	15	6
TABLE ACCESS	CUSTOMER	BY INDEX ROWID	15	2
INDEX	SYS_C00138605	FULL SCAN	15	1
SORT		JOIN	15	4
Access Predicates				
CUSTOMER.CUST_ID=ORDERS.CUST_ID				
Filter Predicates				
CUSTOMER.CUST_ID=ORDERS.CUST_ID				
TABLE ACCESS	ORDERS	FULL	15	3
INDEX	SYS_C00138574	UNIQUE SCAN		
Access Predicates				
EMPLOYEE.EMPLOYEE_ID=ORDERS.EMPLOYEE_ID				
TABLE ACCESS	EMPLOYEE	BY INDEX ROWID	1	3
TABLE ACCESS	EMPLOYEE	FULL	15	3

Query 2:

We retrieved the delivery details of the orders with payment_status as "successful" using sub query operation.

```sql
select delivery_name,delivery_address1,delivery_address2,delivery_mobile
From delivery
where order_id IN
( select order_id from payment where payment_status='successful');
```

Output:

DELIVERY_NAME	DELIVERY_ADDRESS1	DELIVERY_ADDRESS2	DELIVERY_MOBILE
george	22898 louis brown	san francisco, california	8529637412
johnson	55683 revon	north pole, alaska	9505705153
varun	90876 diego rd	stockton, california	7539514563
lohit	909132 kimson	san diego, california	9632587411
kareem	33282 jose aldos	sacramento, california	7896541236
sushmita	12012 fortson	peoria, illunois	6932587456
kajal	33817 downton	chicago , Illunois	7412586412
kareena	90893 chickson	porstborrow, california	7485962132
samantha	22656 zeipod	elgin , illunois	9864712365
manish	33647 skipper rd	chicago , Illunois	8138943901
rashmika	15678 juniper	edwardsville, illunois	7485963236
priyanka	21231 jugad	deeno, arizona	9587412536

Script Output × | Explain Plan × | Query Result ×

SQL | 0.207 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			15	9
HASH JOIN			15	9
Access Predicates				
EMPLOYEE.EMPLOYEE_ID=ORDERS.EMPLOYEE_ID				
NESTED LOOPS			15	9
NESTED LOOPS				
STATISTICS COLLECTOR				
MERGE JOIN			15	6
TABLE ACCESS	CUSTOMER	BY INDEX ROWID	15	2
INDEX	SYS_C00138605	FULL SCAN	15	1
SORT		JOIN	15	4
Access Predicates				
CUSTOMER.CUST_ID=ORDERS.CUST_ID				
Filter Predicates				
CUSTOMER.CUST_ID=ORDERS.CUST_ID				
TABLE ACCESS	ORDERS	FULL	15	3
INDEX	SYS_C00138574	UNIQUE SCAN		
Access Predicates				
EMPLOYEE.EMPLOYEE_ID=ORDERS.EMPLOYEE_ID				
TABLE ACCESS	EMPLOYEE	BY INDEX ROWID	1	3
TABLE ACCESS	EMPLOYEE	FULL	15	3

Query 3:

We retrieved the payment details of the orders with payment_status as "fail" using sub query operation.

```
select payment_id,order_id,payment_type,delivery_price
From payment
where order_id IN
( select order_id from payment where payment_status='fail');
```

Output:

Script Output × Query Result ×

SQL | All Rows Fetched: 3 in 0.027 seconds

	PAYMENT_ID	ORDER_ID	PAYMENT_TYPE	DELIVERY_PRICE
1	pmy002	ord009	cash	45
2	pmy005	ord002	cash	49
3	pmy008	ord007	cash	18

Script Output × Query Result × Explain Plan ×

SQL | 0.233 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
CREATE INDEX STATEMENT			15	2
INDEX BUILD	ORDERS_INDEX	NON UNIQUE		
BITMAP COMPACTION				
SORT		CREATE INDEX	15	
BITMAP CONSTRUCTION				
TABLE ACCESS	ORDERS	FULL	15	1

```
Other XML
  {info}
    info type="db_version"
      12.1.0.2
    info type="parse_schema"
    "SQL327"
    info type="plan_hash_full"
      1739275577
    info type="plan_hash"
      2932472514
    info type="plan_hash_2"
      1739275577
    info type="index_size" note="y"
    65536
```

Performance Tuning:

Creating Indexes to improve the performance:

CREATE BITMAP INDEX customer_index

ON customer(cust_id, cust_mobile);

CREATE BITMAP INDEX delivery_index

ON delivery(delivery_id, order_id);

CREATE BITMAP INDEX employee_index

ON employee(employee_id, login_id);

CREATE BITMAP INDEX item_index

ON item(item_id, item_type);

CREATE BITMAP INDEX orders_index

ON orders(order_id, tracking_id);

CREATE BITMAP INDEX payment_index

ON payment(payment_id, order_id);

CREATE BITMAP INDEX tracking_index

ON tracking(tracking_id, delivery_id);

<u>Post Indexing Query 1 Output:</u>

Script Output | Explain Plan | Query Result

SQL | 0.207 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			15	9
HASH JOIN			15	9
Access Predicates				
EMPLOYEE.EMPLOYEE_ID=ORDERS.EMPLOYEE_ID				
NESTED LOOPS			15	9
NESTED LOOPS				
STATISTICS COLLECTOR				
MERGE JOIN			15	6
TABLE ACCESS	CUSTOMER	BY INDEX ROWID	15	2
INDEX	SYS_C00138605	FULL SCAN	15	1
SORT		JOIN	15	4
Access Predicates				
CUSTOMER.CUST_ID=ORDERS.CUST_ID				
Filter Predicates				
CUSTOMER.CUST_ID=ORDERS.CUST_ID				
TABLE ACCESS	ORDERS	FULL	15	3
INDEX	SYS_C00138574	UNIQUE SCAN		
Access Predicates				
EMPLOYEE.EMPLOYEE_ID=ORDERS.EMPLOYEE_ID				
TABLE ACCESS	EMPLOYEE	BY INDEX ROWID	1	3
TABLE ACCESS	EMPLOYEE	FULL	15	3

Post Indexing Query 2 Output:

Script Output | Explain Plan | Query Result

SQL | 0.207 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			15	9
HASH JOIN			15	9
Access Predicates				
EMPLOYEE.EMPLOYEE_ID=ORDERS.EMPLOYEE_ID				
NESTED LOOPS			15	9
NESTED LOOPS				
STATISTICS COLLECTOR				
MERGE JOIN			15	6
TABLE ACCESS	CUSTOMER	BY INDEX ROWID	15	2
INDEX	SYS_C00138605	FULL SCAN	15	1
SORT		JOIN	15	4
Access Predicates				
CUSTOMER.CUST_ID=ORDERS.CUST_ID				
Filter Predicates				
CUSTOMER.CUST_ID=ORDERS.CUST_ID				
TABLE ACCESS	ORDERS	FULL	15	3
INDEX	SYS_C00138574	UNIQUE SCAN		
Access Predicates				
EMPLOYEE.EMPLOYEE_ID=ORDERS.EMPLOYEE_ID				
TABLE ACCESS	EMPLOYEE	BY INDEX ROWID	1	3
TABLE ACCESS	EMPLOYEE	FULL	15	3

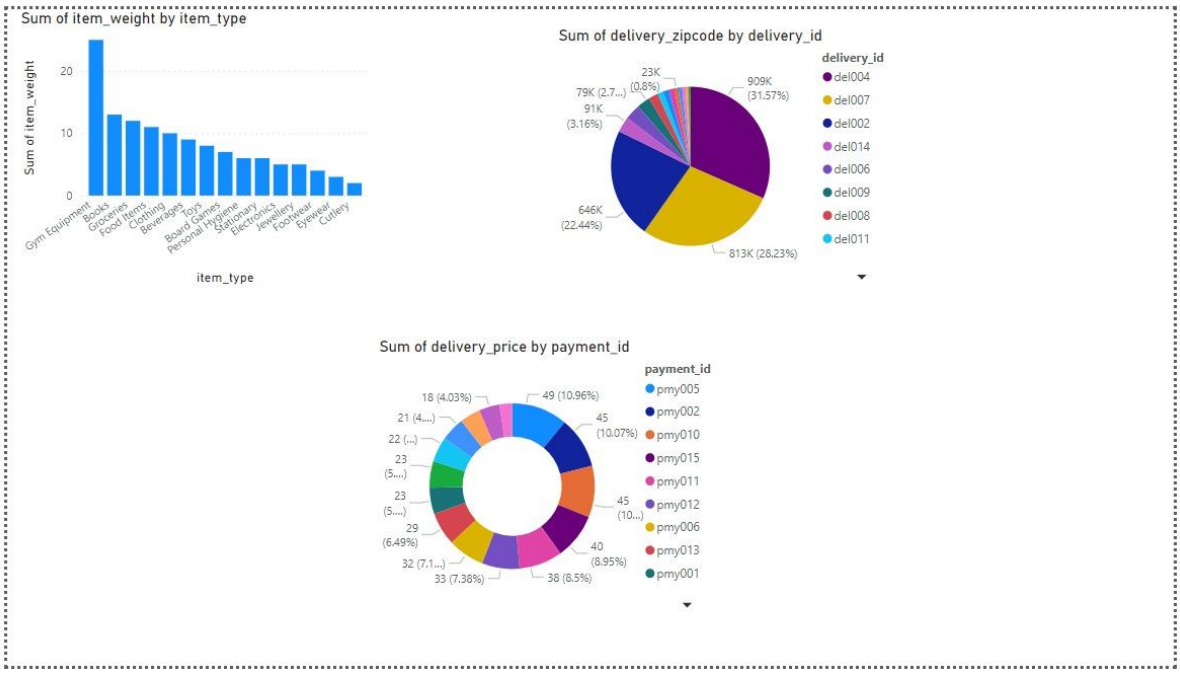
Post Indexing Query 3 Output:

Script Output | Explain Plan | Query Result

SQL | 0.207 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			15	9
HASH JOIN			15	9
Access Predicates				
EMPLOYEE.EMPLOYEE_ID=ORDERS.EMPLOYEE_ID				
NESTED LOOPS			15	9
NESTED LOOPS				
STATISTICS COLLECTOR				
MERGE JOIN			15	6
TABLE ACCESS	CUSTOMER	BY INDEX ROWID	15	2
INDEX	SYS_C00138605	FULL SCAN	15	1
SORT		JOIN	15	4
Access Predicates				
CUSTOMER.CUST_ID=ORDERS.CUST_ID				
Filter Predicates				
CUSTOMER.CUST_ID=ORDERS.CUST_ID				
TABLE ACCESS	ORDERS	FULL	15	3
INDEX	SYS_C00138574	UNIQUE SCAN		
Access Predicates				
EMPLOYEE.EMPLOYEE_ID=ORDERS.EMPLOYEE_ID				
TABLE ACCESS	EMPLOYEE	BY INDEX ROWID	1	3
TABLE ACCESS	EMPLOYEE	FULL	15	3

Visualised the data using power BI:


Sum of item_weight by item_type
Sum of item_weight
20
10
0
Gym Equipment
Books
Groceries
Food Items
Clothing
Beverages
Toys
Board Games
Personal Hygiene
Stationary
Electronics
Jewellery
Footwear
Eyewear
Cutlery
item_type
Sum of delivery_zipcode by delivery_id
delivery_id
del004
del007
del002
del014
del006
del009
del008
del011
909K (31.57%)
813K (28.23%)
646K (22.44%)
91K (3.16%)
79K (2.7...)
23K (0.8%)
Sum of delivery_price by payment_id
payment_id
pmy005
pmy002
pmy010
pmy015
pmy011
pmy012
pmy006
pmy013
pmy001
49 (10.96%)
45 (10.07%)
45 (10...)
40 (8.95%)
38 (8.5%)
33 (7.38%)
32 (7.1...)
29 (6.49%)
23 (5....)
23 (5....)
22 (...)
21 (4....)
18 (4.03%)