



AMERICAN INTERNATIONAL UNIVERSITY BANGLADESH

COURSE: INTRODUCTION TO DATABASE

PROJECT TITLE: E-COMMERCE MANAGEMENT SYSTEM .

GROUP NUMBER: 3

SEMESTER: FALL

COURSE TEACHER: SIFAT RAHMAN AHONA

SECTION: L

ID	NAME
19-41766-3	Saidul Mursalin Khan
19-41749-3	Md Abu Sufian
19-41764-3	Sumit Kanti Sarker
20-41973-1	Abzana Sultan ira

Table of Contents

Introduction.....	3
Case Study	4
ER Diagram.....	5
Normalization	6
Total Table.....	17
Final Table.....	18
Schema Diagram	19
Table Creation.....	20
Data Insertion:.....	37
Query Writing.....	54
Sequence.....	65
Relational Algebra.....	69
Conclusion.....	70

Introduction

E-commerce Management System is an application, refers to the buying and selling of goods or services online, and the transfer of money and data to execute these transactions. For the peoples, it is making life easier. For the management point of view, the admin will able to control the E-commerce system by having all the reports to hand and able to see the records of each customers.

This application will help the Ecommerce to do all functionalities more accurately and faster way by online. E-commerce Management System is often used to refer to the sale of physical products online and improves efficiency of commercial activities. The application can help products ordering from different category and brands and also there are many more functionalities like-

- To store customer information
- Control order and services
- Payments
- Add or remove products based on categories and brands
- Helps admin to control each part of the E-commerce system

The main goal is to maintain the E-commerce system's functions in an effective and accurate manner. This system will help orders to maintain day to day records.

Case Study

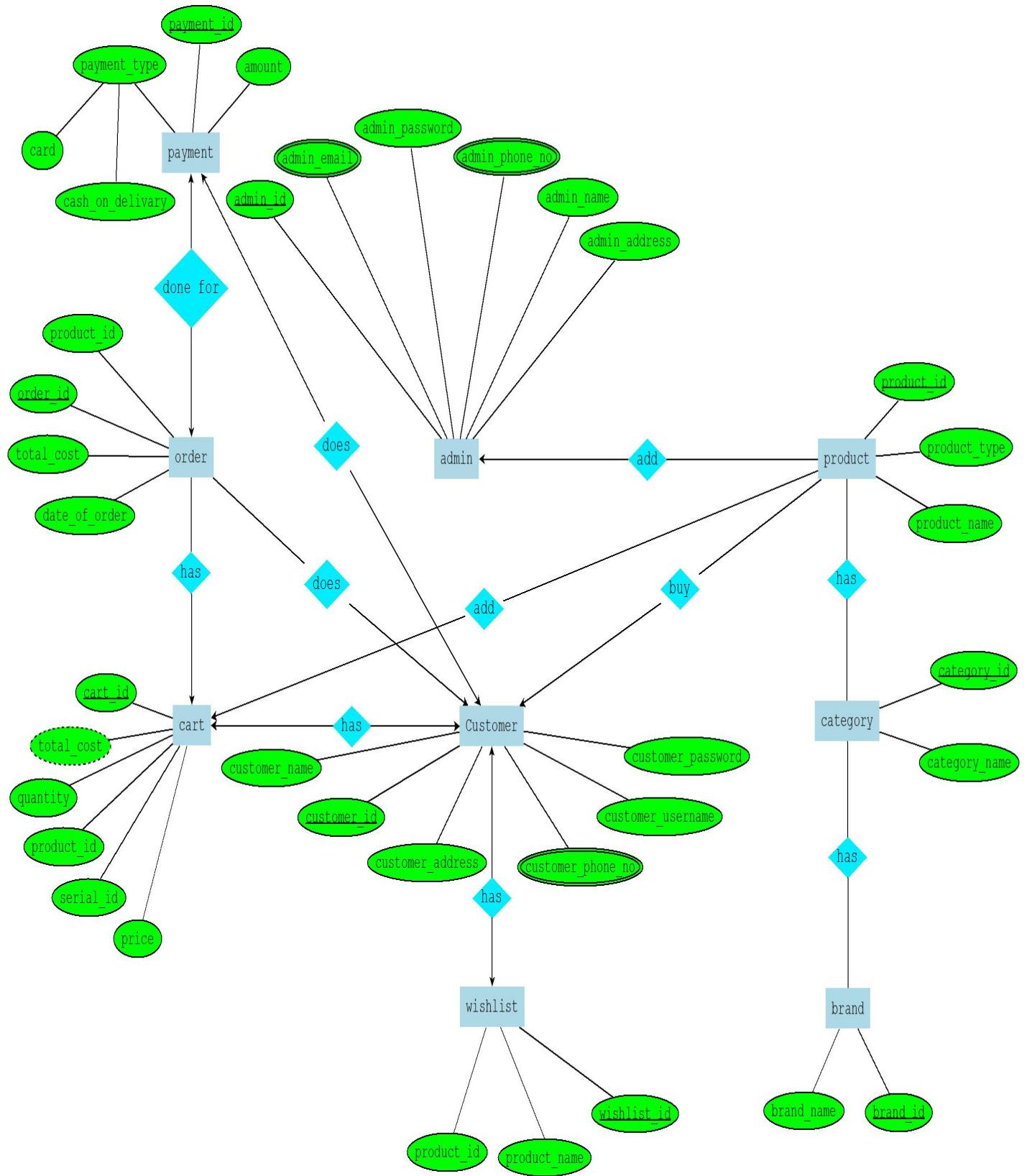
Title: E-commerce management system

Ecommerce, also known as electronic commerce or internet commerce, refers to the buying and selling of goods or services using the internet, and the transfer of money and data to execute these transactions. Ecommerce is often used to refer to the sale of physical products online, but it can also describe any kind of commercial transaction that is facilitated through the internet.

In our e-commerce management system there has a system administrator who can control the system. The administrator's name, email, phone no, address, username and password is stored in the system. The system administrator can add and remove products. There has many types of products. The products are defined with name, id and type. Those products has many categories. The categories has different category id and category name. These category also has different brands and those are identified by brand id and brand name. Customer's name, address, phone number, email, username, password and customer id are stored in the system. Customers are identified with customer id. A customer can login to the system with username and password. Customer can add a product to the wish list in order to buy the product later. Wish list need wish list id, product id and product name. Customer can add any product to the cart. Cart has product quantity, total cost, product id, serial number, per product price and cart id. Customer has an order option from where he/she can order a product from the cart by doing the payment. Order has order id, date of order, product id and total cost of the product. The payment option has amount, payment type (card and cash on delivery) and payment id. After the payment the order will processed.

That is how our e-commerce management system will work.

ER Diagram



Normalization

admin add product

UNF :

add(admin_id, admin_email, admin_password, admin_phone_no, admin_name, admin_address, product_id, product_type, product_name)

1NF:

admin_email, admin_phone_no multivalued attribute.

2NF:

1. admin_id, admin_email, admin_password, admin_phone_no, admin_name, admin_address
2. product_id, product_type, product_name, **admin_id**

3NF:

1. admin_id, admin_email, admin_password, admin_phone_no, admin_name, admin_address
2. product_id, **p_id**, **admin_id**
3. p_id, product_type, product_name

Table:

1. admin_id, admin_password, admin_phone_no, admin_name, **address_id**
2. address_id, admin_address, admin_email.
3. product_id, product_type, product_name, **admin_id**.

product has category

UNF :

has(product_id, product_type, product_name, category_id, category_name)

1NF:

No multivalued attribute.

2NF:

1. product_id, product_type, product_name
2. category_id, category_name
3. **product_id, category_id**

3NF:

1. product_id, product_type, product_name
2. category_id, category_name
3. **product_id, category_id**

No transitive dependency.

Table:

1. product_id, product_type, product_name
2. category_id, category_name
3. **product_id, category_id**

category has brand

UNF :

has(category_id, category_name, brand_id, brand_name)

1NF:

No multivalued attribute.

2NF:

1. category_id, category_name
2. brand_id, brand_name
3. **category_id, brand_id**

3NF:

1. category_id, category_name
2. brand_id, brand_name
3. **category_id, brand_id**

No transitive dependency.

Table:

1. category_id, category_name
2. brand_id, brand_name
3. **category_id, brand_id**

customer has wishlist

UNF :

has(customer_id, customer_name, customer_address, customer_phone_no,
customer_username, customer_password, wishlist_id, customer_id, product_id,
product_name)

1NF:

customer_phone_no multivalued attribute.

2NF:

1. customer_id, customer_name, customer_address, customer_phone_no,
customer_username, customer_password.
2. wishlist_id, product_id, **customer_id**.

3NF:

1. customer_id, customer_name, customer_address, customer_phone_no,
customer_username, customer_password.
2. wishlist_id, product_id, **customer_id**.

No transitive dependency.

Table:

1. customer_id, customer_name, customer_address, customer_phone_no,
customer_username, customer_password.
2. wishlist_id, product_id, **customer_id**.

customer buy product

UNF :

`buy(customer_id, customer_name, customer_address, customer_phone_no,
customer_username, customer_password, product_id, product_type, product_name)`

1NF:

customer_phone_no multivalued attribute.

2NF:

1. customer_id, customer_name, customer_address, customer_phone_no,
customer_username, customer_password
2. product_id, product_type, product_name, **customer_id**

3NF:

1. customer_id, customer_name, customer_address, customer_phone_no,
customer_username, customer_password
2. product_id, product_type, product_name, **customer_id**.

No transitive dependency.

Table:

1. customer_id, customer_name, customer_address, customer_phone_no,
customer_username, customer_password
2. product_id, product_type, product_name, **customer_id**.

customer has cart

UNF :

has(customer_id, customer_name, customer_address, customer_phone_no,
customer_username, customer_password, cart_id, total_cost, quantity, product_id,
serial_id, price)

1NF:

customer_phone_no multivalued attribute.

2NF:

1. customer_id, customer_name, customer_address, customer_phone_no,
customer_username, customer_password
2. cart_id, total_cost, quantity, product_id, serial_id, price, **customer_id**

3NF:

1. customer_id, customer_name, customer_address, customer_phone_no,
customer_username, customer_password
2. cart_id, product_id, serial_id, **customer_id**, **pq_id**
3. pq_id, price, quantity, total_cost

Table:

1. customer_id, customer_name, customer_address, customer_phone_no,
customer_username, customer_password
2. cart_id, product_id, serial_id, **customer_id**, **pq_id**
3. pq_id, price, quantity, total_cost.

customer does payment

UNF :

does(customer_id, customer_name, customer_address, customer_phone_no, customer_username, customer_password, payment_id, card, cash_on_delivary, amount)

1NF:

customer_phone_no multivalued attribute.

2NF:

1. customer_id, customer_name, customer_address, customer_phone_no, customer_username, customer_password
2. payment_id, card, cash_on_delivary, amount, **customer_id**

3NF:

1. customer_id, customer_name, customer_address, customer_phone_no, customer_username, customer_password
2. payment_id, card, cash_on_delivary, amount, **customer_id**

No transitive dependency.

Table:

1. customer_id, customer_name, customer_address, customer_phone_no, customer_username, customer_password
2. payment_id, card, cash_on_delivary, amount, **customer_id**

customer does order

UNF :

does(customer_id, customer_name, customer_address, customer_phone_no,
customer_username, customer_password, order_id, total_cost, product_id,
date_of_order)

1NF:

customer_phone_no multivalued attribute.

2NF:

1. customer_id, customer_name, customer_address, customer_phone_no,
customer_username, customer_password
2. order_id, total_cost, product_id, date_of_order, **customer_id**

3NF:

1. customer_id, customer_name, customer_address, customer_phone_no,
customer_username, customer_password
2. order_id, total_cost, product_id, date_of_order, **customer_id**

No transitive dependency.

Table:

1. customer_id, customer_name, customer_address, customer_phone_no,
customer_username, customer_password
2. order_id, total_cost, product_id, date_of_order, **customer_id**

product add cart

UNF :

add(product_id, product_type, product_name, cart_id, total_cost, quantity, product_id, serial_id, price)

1NF:

No multivalued attribute.

2NF:

1. product_id, product_type, product_name, **cart_id**
2. cart_id, total_cost, quantity, product_id, serial_id, price

3NF:

1. product_id, product_type, product_name, **cart_id**
2. cart_id, product_id, serial_id, **pq_id**
3. pq_id, total_cost, quantity, price

Table:

1. product_id, product_type, product_name, **cart_id**
2. cart_id, product_id, serial_id, **pq_id**
3. pq_id, total_cost, quantity, price.

cart has order

UNF :

`has(cart_id, total_cost, quantity, product_id, serial_id, price, order_id, total_cost, product_id, date_of_order)`

1NF:

No multivalued attribute.

2NF:

1. cart_id, total_cost, quantity, product_id, serial_id, price
2. order_id, total_cost, product_id, date_of_order, **cart_id**

3NF:

1. cart_id, product_id, serial_id, **pq_id**
2. order_id, total_cost, product_id, date_of_order, **cart_id**
3. pq_id, price, quantity, total_cost

Table:

1. cart_id, product_id, serial_id, **pq_id**
2. order_id, total_cost, product_id, date_of_order, **cart_id**
3. pq_id, price, quantity, total_cost.

order done for payment

UNF :

done for(order_id, total_cost, product_id, date_of_order, payment_id, card, cash_on_delivery, amount)

1NF:

No multivalued attribute.

2NF:

1. order_id, total_cost, product_id, date_of_order
2. payment_id, card, cash_on_delivery, amount, **order_id**

3NF:

1. order_id, total_cost, product_id, date_of_order
2. payment_id, card, cash_on_delivery, amount, **order_id**

No transitive dependency.

Table:

1. order_id, total_cost, product_id, date_of_order
2. payment_id, card, cash_on_delivery, amount, **order_id**

Total Table

1. admin_id, admin_password, admin_phone_no, admin_name, **address_id**
2. address_id, admin_address, admin_email
3. product_id, product_type, product_name, **admin_id**
4. product_id, product_type, product_name
5. category_id, category_name
6. **product_id**, category_id
7. category_id, category_name
8. brand_id, brand_name
9. **category_id**, brand_id
10. customer_id, customer_name, customer_address, customer_phone_no,
customer_username, customer_password
11. wishlist_id, product_id, **customer_id**
12. customer_id, customer_name, customer_address, customer_phone_no,
customer_username, customer_password
13. product_id, product_type, product_name, **customer_id**
14. customer_id, customer_name, customer_address, customer_phone_no,
customer_username, customer_password
15. cart_id, product_id, serial_id, **customer_id**, pq_id
16. pq_id, price, quantity, total_cost
17. customer_id, customer_name, customer_address, customer_phone_no,
customer_username, customer_password
18. payment_id, card, cash_on_delivery, amount, **customer_id**
19. customer_id, customer_name, customer_address, customer_phone_no,
customer_username, customer_password
20. order_id, total_cost, product_id, date_of_order, **customer_id**
21. product_id, product_type, product_name, **cart_id**
22. cart_id, product_id, serial_id, pq_id
23. pq_id, total_cost, quantity, price
24. cart_id, product_id, serial_id, pq_id
25. order_id, total_cost, product_id, date_of_order, **cart_id**
26. pq_id, price, quantity, total_cost
27. order_id, total_cost, product_id, date_of_order
28. payment_id, card, cash_on_delivery, amount, **order_id**

Final Table

1. admin_id, admin_password, admin_phone_no, admin_name, **address_id**
2. address_id, admin_address, admin_email
3. product_id, product_type, product_name, **admin_id**
4. category_id, category_name
5. **product_id, category_id**
6. brand_id, brand_name
7. **category_id, brand_id**
8. customer_id, customer_name, customer_address, customer_phone_no, customer_username, customer_password
9. wishlist_id, product_id, **customer_id**
10. product_id, product_type, product_name, **customer_id**
11. cart_id, product_id, serial_id, **customer_id, pq_id**
12. pq_id, price, quantity, total_cost
13. payment_id, card, cash_on_delivery, amount, **customer_id**
14. order_id, total_cost, product_id, date_of_order, **customer_id**
15. product_id, product_type, product_name, **cart_id**
16. order_id, total_cost, product_id, date_of_order, **cart_id**
17. payment_id, card, cash_on_delivery, amount, **order_id**

Schema Diagram

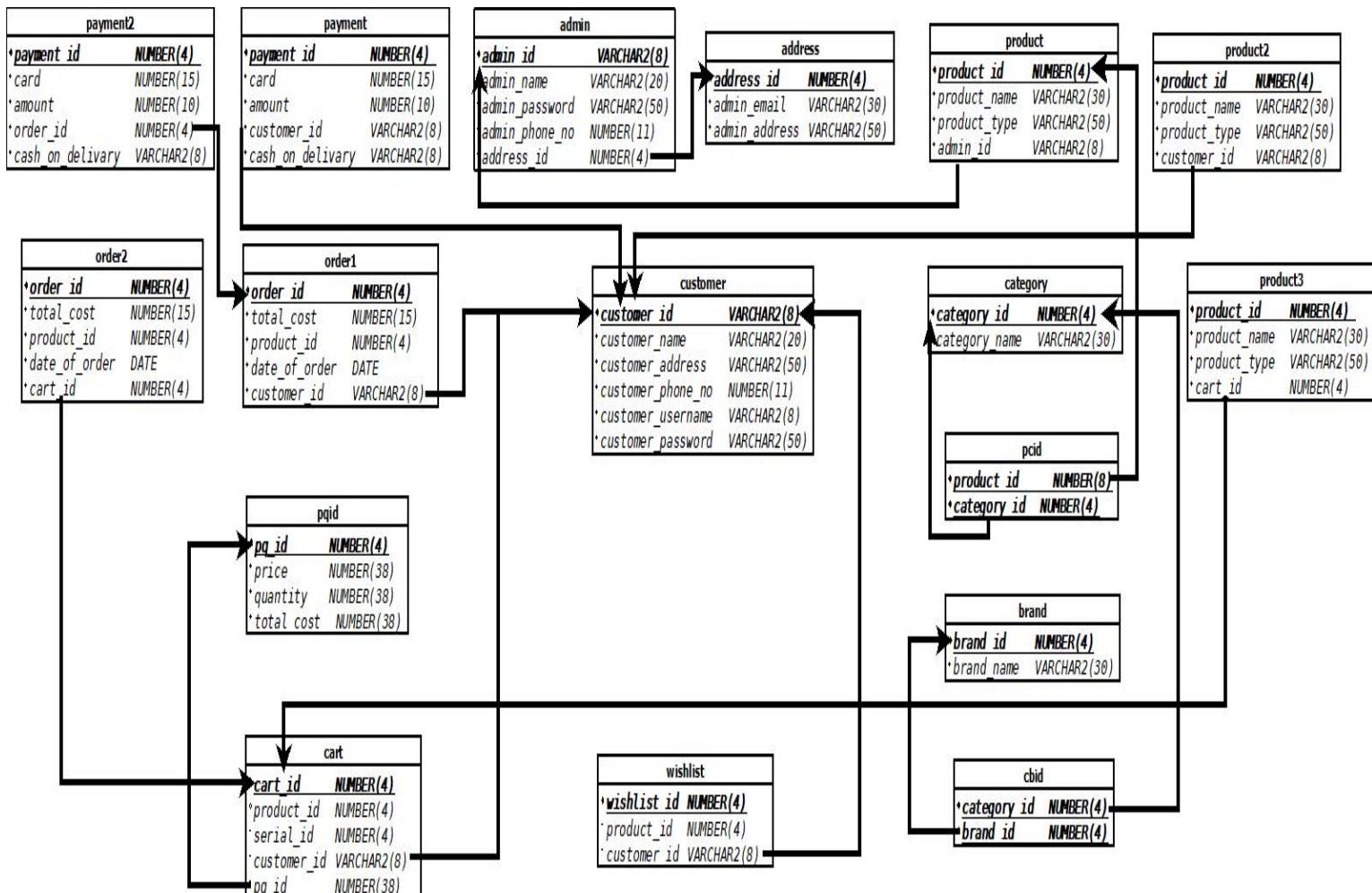


Table Creation

Table : admin

```
CREATE TABLE admin
(
admin_id VARCHAR2(8) NOT NULL,
admin_name VARCHAR2(20) NOT NULL,
admin_password VARCHAR2(50) NOT NULL,
admin_phone_no NUMBER(11) NOT NULL,
address_id NUMBER(4) NOT NULL,
CONSTRAINT admin_PK PRIMARY KEY(admin_id),
CONSTRAINT admin_FK FOREIGN KEY(address_id) REFERENCES
address(address_id)
);
```

DESCRIBE admin;

The screenshot shows the Oracle Database Express Edition interface. The SQL command for creating the 'admin' table is entered in the main editor area. The 'DESCRIBE admin;' command is also present at the bottom of the editor. The results tab is selected, displaying the table structure with columns: ADMIN, ADMIN ID, ADMIN NAME, ADMIN PASSWORD, ADMIN PHONE NO, and ADDRESS ID. The primary key is ADMIN ID. The table has 5 rows of data. At the bottom right, the status bar shows 'Application Express 2.1.0.0.39' and 'Copyright © 1999, 2006, Oracle. All rights reserved.'

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ADMIN	ADMIN ID	Varchar2	8	-	-	1	-	-	-
	ADMIN NAME	Varchar2	20	-	-	-	-	-	-
	ADMIN PASSWORD	Varchar2	50	-	-	-	-	-	-
	ADMIN PHONE NO	Number	-	11	0	-	-	-	-
	ADDRESS ID	Number	-	4	0	-	-	-	-

Table : address

```
CREATE TABLE address
(
address_id NUMBER(4) NOT NULL,
admin_email VARCHAR2(30) UNIQUE,
admin_address VARCHAR2(50) NOT NULL,
CONSTRAINT address_PK PRIMARY KEY(address_id)
);
```

```
DESCRIBE address;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL editor contains the following code:

```
CREATE TABLE address
(
address_id NUMBER(4) NOT NULL,
admin_email VARCHAR2(30) UNIQUE,
admin_address VARCHAR2(50) NOT NULL,
CONSTRAINT address_PK PRIMARY KEY(address_id)
);

DESCRIBE address;
```

Below the editor, the results section displays the table structure:

Object Type	Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TABLE	ADDRESS	ADDRESS_ID	Number	4	0	-	1	-	-	-
		ADMIN_EMAIL	Varchar2	30	-	-	-	✓	-	-
		ADMIN_ADDRESS	Varchar2	50	-	-	-	-	-	-

At the bottom, the status bar shows "Language: en-us" and "Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved."

Table : product

```
CREATE TABLE product
(
product_id NUMBER(4) NOT NULL,
product_name VARCHAR2(30) UNIQUE,
product_type VARCHAR2(50) NOT NULL,
admin_id VARCHAR2(8) NOT NULL,
CONSTRAINT product_PK PRIMARY KEY(product_id),
CONSTRAINT product_FK FOREIGN KEY(admin_id) REFERENCES
admin(admin_id)
);
```

```
DESCRIBE product;
```

ORACLE Database Express Edition

User: ADMIN

Home > SQL > SQL Commands

Autocommit Display 100000

```
CREATE TABLE product
(
product_id NUMBER(4) NOT NULL,
product_name VARCHAR2(30) UNIQUE,
product_type VARCHAR2(50) NOT NULL,
admin_id VARCHAR2(8) NOT NULL,
CONSTRAINT product_PK PRIMARY KEY(product_id),
CONSTRAINT product_FK FOREIGN KEY(admin_id) REFERENCES admin(admin_id)
);

DESCRIBE product;
```

Results Explain Describe Saved SQL History

Object Type TABLE Object PRODUCT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PRODUCT	PRODUCT_ID	Number	-	4	0	1	-	-	-
	PRODUCT_NAME	Varchar2	30	-	-	-	✓	-	-
	PRODUCT_TYPE	Varchar2	50	-	-	-	-	-	-
	ADMIN_ID	Varchar2	8	-	-	-	-	-	-

1 - 4

Application Express 2.1.0.00.39
Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved.

Table : category

```
CREATE TABLE category
(
category_id NUMBER(4) NOT NULL,
category_name VARCHAR2(30) UNIQUE,
CONSTRAINT category_PK PRIMARY KEY(category_id)
);
```

```
DESCRIBE category;
```

The screenshot shows the Oracle Database Express Edition interface. At the top, it says "ORACLE Database Express Edition". Below that, the user is listed as "User: ADMIN". The navigation bar includes "Home", "Logout", and "Help". The main area has tabs for "SQL Commands" and "Object Types". Under "SQL Commands", there is a text input field containing the SQL code for creating the "category" table and its description. Buttons for "Save" and "Run" are at the top right of the input field. Under "Object Types", there is a table titled "TABLE Object CATEGORY". The table has columns: Table, Column, Data Type, Length, Precision, Scale, Primary Key, Nullable, Default, and Comment. It lists two columns for the "CATEGORY" table: "CATEGORY_ID" (Number, 4, 0, 1, Primary Key, Nullable) and "CATEGORY_NAME" (Varchar2, 30, -). A cursor arrow points to the "Primary Key" column of the "CATEGORY_NAME" row. At the bottom of the page, there are links for "Results", "Explain", "Describe", "Saved SQL", and "History". The footer includes "Application Express 2.1.0.00.39", "Language: en-us", and "Copyright © 1999, 2006, Oracle. All rights reserved."

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CATEGORY	CATEGORY_ID	Number	-	4	0	1	-	-	-
CATEGORY	CATEGORY_NAME	Varchar2	30	-	-	-	✓	-	-

Table : pcid

```
CREATE TABLE pcid (
product_id NUMBER(8),
category_id NUMBER(4),
CONSTRAINT pcid_PK PRIMARY KEY(product_id, category_id),
CONSTRAINT pcid_p_FK FOREIGN KEY(product_id) REFERENCES
product(product_id),
CONSTRAINT pcid_c_FK FOREIGN KEY(category_id) REFERENCES
category(category_id)
);
```

```
DESCRIBE pcid;
```

The screenshot shows the Oracle Database Express Edition interface. The top navigation bar includes 'Home', 'Logout', and 'Help' buttons. The main area displays the SQL command for creating the 'pcid' table, followed by the 'DESCRIBE pcid;' command. Below the SQL input field is a results grid showing the table structure.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
POID	PRODUCT_ID	Number	-	8	0	1	-	-	-
	CATEGORY_ID	Number	-	4	0	2	-	-	-
							1-2		

Application Express 2.1.0.00.39

Copyright © 1999, 2006, Oracle. All rights reserved.

Table : brand

```
CREATE TABLE brand
(
brand_id NUMBER(4) NOT NULL,
brand_name VARCHAR2(30) UNIQUE,
CONSTRAINT brand_PK PRIMARY KEY(brand_id)
);

DESCRIBE brand;
```

The screenshot shows the Oracle Database Express Edition interface. The top navigation bar includes links for Home, Logout, and Help. The main area displays the SQL command window with the following content:

```
CREATE TABLE brand
(
brand_id NUMBER(4) NOT NULL,
brand_name VARCHAR2(30) UNIQUE,
CONSTRAINT brand_PK PRIMARY KEY(brand_id)
);

DESCRIBE brand;
```

Below the command window, the results section shows the table structure:

Object Type	TABLE Object BRAND								
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BRAND	BRAND_ID	Number	-	4	0	1	-	-	-
	BRAND_NAME	Varchar2	30	-	-	-	✓	-	-

At the bottom of the page, there are footer links for Application Express 2.1 0 00.39, Copyright © 1999, 2006, Oracle. All rights reserved., and Language: en-us.

Table : cbid

```
CREATE TABLE cbid
(
category_id NUMBER(4),
brand_id NUMBER(4),
CONSTRAINT cbid_PK PRIMARY KEY(category_id, brand_id),
CONSTRAINT cbid_c_FK FOREIGN KEY(category_id) REFERENCES
category(category_id),
CONSTRAINT cbid_b_FK FOREIGN KEY(brand_id) REFERENCES
brand(brand_id)
);

DESCRIBE cbid;
```

ORACLE Database Express Edition

User: ADMIN

Home > SQL > SQL Commands

Autocommit Save Run

```
CREATE TABLE cbid
(
category_id NUMBER(4),
brand_id NUMBER(4),
CONSTRAINT cbid_PK PRIMARY KEY(category_id, brand_id),
CONSTRAINT cbid_c_FK FOREIGN KEY(category_id) REFERENCES category(category_id),
CONSTRAINT cbid_b_FK FOREIGN KEY(brand_id) REFERENCES brand(brand_id)
);
DESCRIBE cbid;
```

Results Explain Describe Saved SQL History

Object Type TABLE Object CBID

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CBID	CATEGORY_ID	Number	-	4	0	1	-	-	-
	BRAND_ID	Number	-	4	0	2	-	-	-

1-2

Application Express 2.1.0.0.39
Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved.

Table : customer

```
CREATE TABLE customer
(
customer_id VARCHAR2(8) NOT NULL,
customer_name VARCHAR2(20) NOT NULL,
customer_address VARCHAR2(50) NOT NULL,
customer_phone_no NUMBER(11) NOT NULL,
customer_username VARCHAR2(8) NOT NULL,
customer_password VARCHAR2(50) NOT NULL,
CONSTRAINT customer_PK PRIMARY KEY(customer_id),
CONSTRAINT customer_unique UNIQUE (customer_id, customer_name)
);
```

```
DESCRIBE customer;
```

User: ADMIN

Home > SQL > SQL Commands

Autocommit

```
CREATE TABLE customer
(
customer_id VARCHAR2(8) NOT NULL,
customer_name VARCHAR2(20) NOT NULL,
customer_address VARCHAR2(50) NOT NULL,
customer_phone_no NUMBER(11) NOT NULL,
customer_username VARCHAR2(8) NOT NULL,
customer_password VARCHAR2(50) NOT NULL,
CONSTRAINT customer_PK PRIMARY KEY(customer_id),
CONSTRAINT customer_unique UNIQUE (customer_id, customer_name)
);

DESCRIBE customer;
```

Results Explain Describe Saved SQL History

Object Type TABLE Object CUSTOMER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER	CUSTOMER_ID	Varchar2	8	-	-	1	-	-	-
CUSTOMER	CUSTOMER_NAME	Varchar2	20	-	-	-	-	-	-
CUSTOMER	CUSTOMER_ADDRESS	Varchar2	50	-	-	-	-	-	-
CUSTOMER	CUSTOMER_PHONE_NO	Number	-	11	0	-	-	-	-
CUSTOMER	CUSTOMER_USERNAME	Varchar2	8	-	-	-	-	-	-
CUSTOMER	CUSTOMER_PASSWORD	Varchar2	50	-	-	-	-	-	-

1 - 6

Application Express 2.1.0.0.39
Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved.

Table : wishlist

```
CREATE TABLE wishlist
(
wishlist_id NUMBER(4) NOT NULL,
product_id NUMBER(4) NOT NULL,
customer_id VARCHAR2(8) NOT NULL,
CONSTRAINT wishlist_PK PRIMARY KEY(wishlist_id),
CONSTRAINT wishlist_unique UNIQUE (wishlist_id, product_id, customer_id)
);

DESCRIBE wishlist;
```

The screenshot shows the Oracle Database Express Edition interface. The top navigation bar includes links for Home, Logout, and Help. The main menu bar has options like User: ADMIN, Home > SQL > SQL Commands. The SQL editor window contains the SQL code for creating the wishlist table and its description. Below the editor, the Results tab is selected, showing the table structure for WISHLIST with three columns: WISHLIST_ID, PRODUCT_ID, and CUSTOMER_ID. At the bottom, there are footer links for Application Express 2.1.0.0.39, Copyright © 1999, 2006, Oracle. All rights reserved., and Language: en-us.

```
CREATE TABLE wishlist
(
wishlist_id NUMBER(4) NOT NULL,
product_id NUMBER(4) NOT NULL,
customer_id VARCHAR2(8) NOT NULL,
CONSTRAINT wishlist_PK PRIMARY KEY(wishlist_id),
CONSTRAINT wishlist_unique UNIQUE (wishlist_id, product_id, customer_id)
);

DESCRIBE wishlist;
```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
WISHLIST	WISHLIST_ID	Number	-	4	0	1	-	-	-
	PRODUCT_ID	Number	-	4	0	-	-	-	-
	CUSTOMER_ID	Varchar2	8	-	-	-	-	-	-

Table : product2

```
CREATE TABLE product2
(
product_id NUMBER(4) NOT NULL,
product_name VARCHAR2(30) UNIQUE,
product_type VARCHAR2(50) NOT NULL,
customer_id VARCHAR2(8) NOT NULL,
CONSTRAINT product2_PK PRIMARY KEY(product_id),
CONSTRAINT product2_FK FOREIGN KEY(customer_id) REFERENCES
customer(customer_id),
CONSTRAINT product2_unique UNIQUE (product_id, product_name,
product_type)
);
```

```
DESCRIBE product2;
```

The screenshot shows the Oracle Database Express Edition interface. The top navigation bar includes links for Home, Logout, and Help. The main area displays the SQL command window with the following content:

```
CREATE TABLE product2
(
product_id NUMBER(4) NOT NULL,
product_name VARCHAR2(30) UNIQUE,
product_type VARCHAR2(50) NOT NULL,
customer_id VARCHAR2(8) NOT NULL,
CONSTRAINT product2_PK PRIMARY KEY(product_id),
CONSTRAINT product2_FK FOREIGN KEY(customer_id) REFERENCES customer(customer_id),
CONSTRAINT product2_unique UNIQUE (product_id, product_name, product_type)
);
DESCRIBE product2;
```

Below the command window, the results section shows the table structure:

Object Type	TABLE Object	PRODUCT2							
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PRODUCT2	PRODUCT_ID	Number	-	4	0	1	-	-	-
	PRODUCT_NAME	Varchar2	30	-	-	-	✓	-	-
	PRODUCT_TYPE	Varchar2	50	-	-	-	-	-	-
	CUSTOMER_ID	Varchar2	8	-	-	-	-	-	-

At the bottom, the status bar indicates "Language: en-us" and "Application Express 2.1.0 00:39".

Table : cart

```
CREATE TABLE cart
(
    cart_id NUMBER(4) PRIMARY KEY,
    product_id NUMBER(4) UNIQUE,
    serial_id NUMBER(4) NOT NULL,
    customer_id VARCHAR2(8) NOT NULL,
    pq_id NUMBER(38),
    CONSTRAINT cart_c_FK FOREIGN KEY(customer_id) REFERENCES
    customer(customer_id),
    CONSTRAINT cart_pq_FK FOREIGN KEY(pq_id) REFERENCES pqid(pq_id),
    CONSTRAINT cart_unique UNIQUE (serial_id, customer_id, cart_id)
);
DESCRIBE cart;
```

The screenshot shows the Oracle Database Express Edition interface. The top navigation bar includes links for Home, Logout, and Help. The main area displays the SQL command for creating the 'cart' table, which includes primary key, unique constraints, and foreign key references to the 'customer' and 'pqid' tables. Below the SQL input field are tabs for Results, Explain, Describe, Saved SQL, and History. The 'Describe' tab is selected, showing the table structure with columns: CART_ID, PRODUCT_ID, SERIAL_ID, CUSTOMER_ID, and PQ_ID. The CUSTOMER_ID column is marked as a Primary Key. At the bottom of the page, there are links for Application Express 2.1.0.00.39, Copyright © 1999, 2006, Oracle. All rights reserved., and Language: en-us.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CART	CART_ID	Number	-	4	0	1	-	-	-
	PRODUCT_ID	Number	-	4	0	-	✓	-	-
	SERIAL_ID	Number	-	4	0	-	-	-	-
	CUSTOMER_ID	Varchar2	8	-	-	-	-	-	-
	PQ_ID	Number	-	38	0	-	✓	-	-

Table : pqid

```
CREATE TABLE pqid
(
    pq_id NUMBER(4) PRIMARY KEY,
    price NUMBER(38) NOT NULL,
    quantity NUMBER(38) NOT NULL,
    total_cost NUMBER(38) NOT NULL
);
```

```
DESCRIBE pqid;
```

The screenshot shows the Oracle Database Express Edition interface. The top navigation bar includes links for Home, Logout, and Help. The main area displays the SQL Commands page. A code editor window contains the SQL statements for creating the pqid table and describing it. Below the code editor is a results grid showing the table structure with four columns: PQ_ID, PRICE, QUANTITY, and TOTAL COST. The bottom status bar indicates the application version (Application Express 2.1.0.00.39), the language (en-us), and copyright information (Copyright © 1999, 2006, Oracle. All rights reserved.).

```
ORACLE Database Express Edition
User: ADMIN
Home > SQL > SQL Commands
Autocommit Display 100000
Save Run
CREATE TABLE pqid
(
    pq_id NUMBER(4) PRIMARY KEY,
    price NUMBER(38) NOT NULL,
    quantity NUMBER(38) NOT NULL,
    total_cost NUMBER(38) NOT NULL
);
DESCRIBE pqid;
```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PQID	PQ_ID	Number	-	4	0	1	-	-	-
	PRICE	Number	-	38	0	-	-	-	-
	QUANTITY	Number	-	38	0	-	-	-	-
	TOTAL COST	Number	-	38	0	-	-	-	-
									1 - 4

Table : payment

```
CREATE TABLE payment
(
payment_id NUMBER(4) NOT NULL,
card NUMBER(15),
amount NUMBER(10) NOT NULL,
customer_id VARCHAR2(8) NOT NULL,
cash_on_delivery VARCHAR2(8) NOT NULL,
CONSTRAINT payment_PK PRIMARY KEY(payment_id),
CONSTRAINT payment_FK FOREIGN KEY(customer_id) REFERENCES
customer(customer_id),
CONSTRAINT payment_unique UNIQUE (payment_id, customer_id)
);
DESCRIBE payment;
```

The screenshot shows the Oracle SQL Developer interface. The top navigation bar includes 'Home', 'Logout', and 'Help'. The user is logged in as 'ADMIN'. The main area shows the SQL command window with the code for creating the 'payment' table and its description. The code uses red highlighting for errors or warnings. Below the code, the 'Results' tab is selected, displaying the table structure with columns: PAYMENT_ID, CARD, AMOUNT, CUSTOMER_ID, and CASH_ON_DELIVERY. The bottom status bar indicates the application version as 'Application Express 2.1.0.0.39' and the language as 'en-us'.

```
CREATE TABLE payment
(
payment_id NUMBER(4) NOT NULL,
card NUMBER(15) UNIQUE,
amount NUMBER(10) NOT NULL,
customer_id VARCHAR2(8) NOT NULL,
cash_on_delivery VARCHAR2(8),
CONSTRAINT payment_PK PRIMARY KEY(payment_id),
CONSTRAINT payment_FK FOREIGN KEY(customer_id) REFERENCES customer(customer_id),
CONSTRAINT payment_unique UNIQUE (payment_id, customer_id)
);

DESCRIBE payment;
```

Object Type	Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PAYMENT	PAYMENT_ID	Number	-	4	0	1	-	-	-	-
	CARD	Number	-	15	0	-	✓	-	-	-
	AMOUNT	Number	-	10	0	-	-	-	-	-
	CUSTOMER_ID	Varchar2	8	-	-	-	-	-	-	-
	CASH ON DELIVERY	Varchar2	8	-	-	-	✓	-	-	-

Table : order1

```
CREATE TABLE order1
(
    order_id NUMBER(4) NOT NULL,
    total_cost NUMBER(15) NOT NULL,
    product_id NUMBER(4) NOT NULL,
    date_of_order DATE,
    customer_id VARCHAR2(8) NOT NULL,
    CONSTRAINT order1_PK PRIMARY KEY(order_id),
    CONSTRAINT order1_FK FOREIGN KEY(customer_id) REFERENCES
    customer(customer_id),
    CONSTRAINT order1_unique UNIQUE (order_id, customer_id, product_id)
);
```

```
DESCRIBE order1;
```

The screenshot shows the Oracle Database Express Edition interface. The top navigation bar includes links for Home, Logout, and Help. The main workspace displays the SQL Commands page. A code editor window contains the SQL code for creating the order1 table, which is highlighted with red underlines. Below the code editor is a command line input field with 'DESCRIBE order1;' typed into it. At the bottom of the page, there are tabs for Results, Explain, Describe, Saved SQL, and History, with 'Describe' being the active tab. A table below shows the object type 'TABLE Object ORDER1' with columns for Table, Column, Data Type, Length, Precision, Scale, Primary Key, Nullable, Default, and Comment. The table data is as follows:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ORDER1	ORDER_ID	Number	-	4	0	1	-	-	-
	TOTAL_COST	Number	-	15	0	-	-	-	-
	PRODUCT_ID	Number	-	4	0	-	-	-	-
	DATE_OF_ORDER	Date	7	-	-	-	✓	-	-
	CUSTOMER_ID	Varchar2	8	-	-	-	-	-	-

1-5

At the bottom right, the footer includes 'Application Express 2.1.0.0.39', 'Language: en-us', and 'Copyright © 1999, 2006, Oracle. All rights reserved.'

Table : product3

```
CREATE TABLE product3
(
product_id NUMBER(4) NOT NULL,
product_name VARCHAR2(30) UNIQUE,
product_type VARCHAR2(50) NOT NULL,
cart_id NUMBER(4) NOT NULL,
CONSTRAINT product3_PK PRIMARY KEY(product_id),
CONSTRAINT product3_FK FOREIGN KEY(cart_id) REFERENCES
cart(cart_id),
CONSTRAINT product3_unique UNIQUE (product_id, product_type, cart_id)
);

DESCRIBE product3;
```

The screenshot shows the Oracle Database Express Edition interface. The top navigation bar includes links for Home, Logout, and Help. The main workspace displays the SQL Commands page. A code editor window contains the SQL script for creating the product3 table, including its columns, constraints, and a DESCRIBE statement at the end. Below the code editor is a results pane showing the table structure with four columns: PRODUCT_ID, PRODUCT_NAME, PRODUCT_TYPE, and CART_ID. The bottom of the screen shows application details like the language (en-us) and copyright information (Copyright © 1999, 2006, Oracle. All rights reserved.).

```
CREATE TABLE product3
(
product_id NUMBER(4) NOT NULL,
product_name VARCHAR2(30) UNIQUE,
product_type VARCHAR2(50) NOT NULL,
cart_id NUMBER(4) NOT NULL,
CONSTRAINT product3_PK PRIMARY KEY(product_id),
CONSTRAINT product3_FK FOREIGN KEY(cart_id) REFERENCES cart(cart_id),
CONSTRAINT product3_unique UNIQUE (product_id, product_type, cart_id)
);

DESCRIBE product3;
```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PRODUCT3	PRODUCT_ID	Number	-	4	0	1	-	-	-
	PRODUCT_NAME	Varchar2	30	-	-	-	✓	-	-
	PRODUCT_TYPE	Varchar2	50	-	-	-	-	-	-
	CART_ID	Number	-	4	0	-	-	-	-

Table : order2

```
CREATE TABLE order2
(
payment_id NUMBER(4) NOT NULL,
card NUMBER(15),
amount NUMBER(10) NOT NULL,
order_id NUMBER(4) NOT NULL,
cash_on_delivery VARCHAR2(8),
CONSTRAINT payment2_PK PRIMARY KEY(payment_id),
CONSTRAINT payment2_FK FOREIGN KEY(order_id) REFERENCES
order1(order_id),
CONSTRAINT payment2_unique UNIQUE (payment_id, order_id)
);
DESCRIBE order2;
```

The screenshot shows the Oracle Database Express Edition interface. The SQL Commands page displays the creation of the order2 table. The table structure includes columns for payment_id (NUMBER(4)), card (NUMBER(15)), amount (NUMBER(10)), order_id (NUMBER(4)), and cash_on_delivery (VARCHAR2(8)). It features a primary key constraint on payment_id and a foreign key constraint on order_id referencing the order1 table. A unique constraint is also applied to the combination of payment_id and order_id. The 'DESCRIBE order2;' command is shown at the bottom of the query window.

User: ADMIN

Home > SQL > SQL Commands

Autocommit Display 100000 Save Run

```
CREATE TABLE order2
(
order_id NUMBER(4) NOT NULL,
total_cost NUMBER(15) NOT NULL,
product_id NUMBER(4) NOT NULL,
date_of_order DATE,
cart_id NUMBER(4) NOT NULL,
CONSTRAINT order2_PK PRIMARY KEY(order_id),
CONSTRAINT order2_FK FOREIGN KEY(cart_id) REFERENCES cart(cart_id),
CONSTRAINT order2_unique UNIQUE (order_id, cart_id, product_id)
);
DESCRIBE order2;
```

Results Explain Describe Saved SQL History

Object Type TABLE Object ORDER2

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ORDER2	ORDER_ID	Number	-	4	0	1	-	-	-
	TOTAL_COST	Number	-	15	0	-	-	-	-
	PRODUCT_ID	Number	-	4	0	-	-	-	-
	DATE_OF_ORDER	Date	7	-	-	-	✓	-	-
	CART_ID	Number	-	4	0	-	-	-	-

1-5

Application Express 2.1.0.00.39
Language: en-us
Copyright © 1999, 2006, Oracle. All rights reserved.

Table : payment2

```
CREATE TABLE payment2
(
payment_id NUMBER(4) NOT NULL,
card NUMBER(15),
amount NUMBER(10) NOT NULL,
order_id VARCHAR2(8) NOT NULL,
cash_on_delivery VARCHAR2(8) NOT NULL,
CONSTRAINT payment2_PK PRIMARY KEY(payment_id),
CONSTRAINT payment2_FK FOREIGN KEY(order_id) REFERENCES
order1(order_id),
CONSTRAINT payment2_unique UNIQUE (payment_id, order_id)
);
```

```
DESCRIBE payment2;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL editor window contains the following code:

```
CREATE TABLE payment2
(
payment_id NUMBER(4) NOT NULL,
card NUMBER(15) UNIQUE,
amount NUMBER(10) NOT NULL,
order_id NUMBER(4) NOT NULL,
cash_on_delivery VARCHAR2(8),
CONSTRAINT payment2_PK PRIMARY KEY(payment_id),
CONSTRAINT payment2_FK FOREIGN KEY(order_id) REFERENCES order1(order_id),
CONSTRAINT payment2_unique UNIQUE (payment_id, order_id)
);
```

Below the code, the command `DESCRIBE payment2;` is entered. The results section displays the table structure:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PAYMENT2	PAYMENT_ID	Number	-	4	0	1	-	-	-
	CARD	Number	-	15	0	-	✓	-	-
	AMOUNT	Number	-	10	0	-	-	-	-
	ORDER_ID	Number	-	4	0	-	-	-	-
	CASH_ON_DELIVERY	Varchar2	8	-	-	-	✓	-	-

At the bottom of the results, it says "1 - 5". The status bar at the bottom right indicates "Application Express 2.1.0.00.39" and "Copyright © 1999, 2006, Oracle. All rights reserved."

Data Insertion:

1. admin

```
INSERT INTO admin(admin_id, admin_name, admin_password, admin_phone_no, address_id) VALUES('admin', 'alex', 'admin1234', '0170000000', 01);
```

```
select * from admin
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL editor contains the following code:

```
INSERT INTO admin(admin_id, admin_name, admin_password, admin_phone_no, address_id) VALUES('admin', 'alex', 'admin1234', '0170000000', 01);
select * from admin;
```

The results section displays the inserted data:

ADMIN_ID	ADMIN_NAME	ADMIN_PASSWORD	ADMIN_PHONE_NO	ADDRESS_ID
admin	alex	admin1234	170000000	1

1 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.00.39

2. address

```
INSERT INTO address (address_id, admin_email, admin_address) VALUES (01, 'admin@gmail.com', 'Dhaka,Bangladesh');
select * from address;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. At the top, there are navigation links for Home, Logout, and Help. Below that, the user is identified as ADMIN. The main area contains a SQL editor window with the following content:

```
Autocommit: checked | Display: 10 | Save | Run
INSERT INTO address (address_id, admin_email, admin_address) VALUES (01, 'admin@gmail.com', 'Dhaka,Bangladesh');
select * from address;
```

Below the editor, there are tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is selected, showing the following table output:

ADDRESS_ID	ADMIN_EMAIL	ADMIN_ADDRESS
1	admin@gmail.com	Dhaka,Bangladesh

1 rows returned in 0.00 seconds | [CSV Export](#)

At the bottom of the interface, there are language and copyright information:

Language: en-us Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.

3. product

```
INSERT INTO product (product_id, product_name, product_type, admin_id) VALUES (001, 'Android Mobile', 'Gadget', 'admin');

INSERT INTO product (product_id, product_name, product_type, admin_id) VALUES (002, 'Core i7 Laptop', 'Gadget', 'admin');

INSERT INTO product (product_id, product_name, product_type, admin_id) VALUES (003, 'Mens Casual Shirt', 'Fashion', 'admin');

INSERT INTO product (product_id, product_name, product_type, admin_id) VALUES (004, 'Mens Formal Pant', 'Fashion', 'admin');

INSERT INTO product (product_id, product_name, product_type, admin_id) VALUES (005, ' Oracle Database 12c Book', 'Stationary', 'admin');

INSERT INTO product (product_id, product_name, product_type, admin_id) VALUES (006, 'Lip Stick', 'Cosmetics', 'admin');

INSERT INTO product (product_id, product_name, product_type, admin_id) VALUES (007, 'Body Lotion', 'Cosmetics', 'admin');

INSERT INTO product (product_id, product_name, product_type, admin_id) VALUES (008, 'Rice', 'Glossary', 'admin');

INSERT INTO product (product_id, product_name, product_type, admin_id) VALUES (009, 'Flour', 'Glossary', 'admin');

INSERT INTO product (product_id, product_name, product_type, admin_id) VALUES (010, 'Soyabean Oil', 'Glossary', 'admin');

select * from product;
```

The screenshot shows the Oracle Database Express Edition interface. The SQL command window contains the previously listed SQL statements for inserting data into the 'product' table and selecting all rows. The results window displays the 10 rows returned, each with a unique product ID, name, type, and admin ID.

PRODUCT_ID	PRODUCT_NAME	PRODUCT_TYPE	ADMIN_ID
10	Soyabean Oil	Glossary	admin
1	Android Mobile	Gadget	admin
2	Core i7 Laptop	Gadget	admin
4	Mens Formal Pant	Fashion	admin
3	Mens Casual Shirt	Fashion	admin
5	Oracle Database 12c Book	Stationary	admin
6	Lip Stick	Cosmetics	admin
7	Body Lotion	Cosmetics	admin
8	Rice	Glossary	admin
9	Flour	Glossary	admin

4. category

```
INSERT INTO category (category_id, category_name) VALUES (011, 'Mobile Phone');

INSERT INTO category (category_id, category_name) VALUES (012, 'Laptop');

INSERT INTO category (category_id, category_name) VALUES (013, 'Dress');

INSERT INTO category (category_id, category_name) VALUES (014, 'Pants');

INSERT INTO category (category_id, category_name) VALUES (015, 'Books');

INSERT INTO category (category_id, category_name) VALUES (016, 'Cosmetics');

INSERT INTO category (category_id, category_name) VALUES (017, 'Cosmetics-liquid');

INSERT INTO category (category_id, category_name) VALUES (018, 'Glossary-paddy');

INSERT INTO category (category_id, category_name) VALUES (019, 'Glossary-meal');

INSERT INTO category (category_id, category_name) VALUES (020, 'Glossary-oil');

select * from category;
```

The screenshot shows the Oracle Database Express Edition interface. The SQL command window contains the same SQL code as above. Below it, the results window displays a table with two columns: CATEGORY_ID and CATEGORY_NAME. The data is as follows:

CATEGORY_ID	CATEGORY_NAME
16	Cosmetics
17	Cosmetics-liquid
18	Glossary-paddy
19	Glossary-meal
20	Glossary-oil
11	Mobile Phone
12	Laptop
13	Dress
14	Pants
15	Books

At the bottom, it says "10 rows returned in 0.00 seconds". The status bar at the bottom right indicates "Application Express 2.1.0.0.39" and "Copyright © 1999, 2008, Oracle. All rights reserved."

5. pcid

```
INSERT INTO pcid (product_id, category_id) VALUES (001, 011);
INSERT INTO pcid (product_id, category_id) VALUES (002, 012);
INSERT INTO pcid (product_id, category_id) VALUES (003, 013);
INSERT INTO pcid (product_id, category_id) VALUES (004, 014);
INSERT INTO pcid (product_id, category_id) VALUES (005, 015);
INSERT INTO pcid (product_id, category_id) VALUES (006, 016);
INSERT INTO pcid (product_id, category_id) VALUES (007, 017);
INSERT INTO pcid (product_id, category_id) VALUES (008, 018);
INSERT INTO pcid (product_id, category_id) VALUES (009, 019);
INSERT INTO pcid (product_id, category_id) VALUES (010, 020);
select * from pcid;
```

The screenshot shows the Oracle Database Express Edition interface. The SQL command window contains the provided SQL script. The results tab shows a table with 10 rows of data, where each row has a PRODUCT_ID from 1 to 10 and a corresponding CATEGORY_ID.

PRODUCT_ID	CATEGORY_ID
1	11
2	12
3	13
4	14
5	15
6	16
7	17
8	18
9	19
10	20

10 rows returned in 0.02 seconds [CSV Export](#)

6. brand

```
INSERT INTO brand (brand_id, brand_name) VALUES (111, 'Samsung');

INSERT INTO brand (brand_id, brand_name) VALUES (112, 'ASUS');

INSERT INTO brand (brand_id, brand_name) VALUES (113, 'Yellow');

INSERT INTO brand (brand_id, brand_name) VALUES (114, 'Cat's Eye');

INSERT INTO brand (brand_id, brand_name) VALUES (115, 'Springer Nature');

INSERT INTO brand (brand_id, brand_name) VALUES (116, 'Lakme');

INSERT INTO brand (brand_id, brand_name) VALUES (117, 'Nevia');

INSERT INTO brand (brand_id, brand_name) VALUES (118, 'Pran');

INSERT INTO brand (brand_id, brand_name) VALUES (119, 'Teer');

INSERT INTO brand (brand_id, brand_name) VALUES (120, 'Fresh');

select * from brand;
```

ORACLE® Database Express Edition

User: ADMIN

Home > SQL > SQL Commands

Autocommit

```
INSERT INTO brand (brand_id, brand_name) VALUES (111, 'Samsung');
INSERT INTO brand (brand_id, brand_name) VALUES (112, 'ASUS');
INSERT INTO brand (brand_id, brand_name) VALUES (113, 'Yellow');
INSERT INTO brand (brand_id, brand_name) VALUES (114, 'Cat's Eye');
INSERT INTO brand (brand_id, brand_name) VALUES (115, 'Springer Nature');
INSERT INTO brand (brand_id, brand_name) VALUES (116, 'Lakme');
INSERT INTO brand (brand_id, brand_name) VALUES (117, 'Nevia');
INSERT INTO brand (brand_id, brand_name) VALUES (118, 'Pran');
INSERT INTO brand (brand_id, brand_name) VALUES (119, 'Teer');
INSERT INTO brand (brand_id, brand_name) VALUES (120, 'Fresh');

select * from brand;
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

BRAND_ID	BRAND_NAME
116	Lakme
117	Nevia
118	Pran
119	Teer
120	Fresh
111	Samsung
112	ASUS
113	Yellow
114	Cat's Eye
115	Springer Nature

10 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2008, Oracle. All rights reserved.

7. cbid

```
INSERT INTO cbid (category_id,brand_id) VALUES (011, 111);
INSERT INTO cbid (category_id,brand_id) VALUES (012, 112);
INSERT INTO cbid (category_id,brand_id) VALUES (013, 113);
INSERT INTO cbid (category_id,brand_id) VALUES (014, 114);
INSERT INTO cbid (category_id,brand_id) VALUES (015, 115);
INSERT INTO cbid (category_id,brand_id) VALUES (016, 116);
INSERT INTO cbid (category_id,brand_id) VALUES (017, 117);
INSERT INTO cbid (category_id,brand_id) VALUES (018, 118);
INSERT INTO cbid (category_id,brand_id) VALUES (019, 119);
INSERT INTO cbid (category_id,brand_id) VALUES (020, 120);
select * from cbid;
```

The screenshot shows the Oracle Database Express Edition interface. The top navigation bar includes links for Home, Logout, and Help. The main area displays the SQL command window with the following content:

```
ORACLE Database Express Edition
User: ADMIN
Home > SQL > SQL Commands
Autocommit Display 10 Save Run
INSERT INTO cbid (category_id,brand_id) VALUES (011, 111);
INSERT INTO cbid (category_id,brand_id) VALUES (012, 112);
INSERT INTO cbid (category_id,brand_id) VALUES (013, 113);
INSERT INTO cbid (category_id,brand_id) VALUES (014, 114);
INSERT INTO cbid (category_id,brand_id) VALUES (015, 115);
INSERT INTO cbid (category_id,brand_id) VALUES (016, 116);
INSERT INTO cbid (category_id,brand_id) VALUES (017, 117);
INSERT INTO cbid (category_id,brand_id) VALUES (018, 118);
INSERT INTO cbid (category_id,brand_id) VALUES (019, 119);
INSERT INTO cbid (category_id,brand_id) VALUES (020, 120);
select * from cbid;
```

Below the command window, the Results tab is selected, showing the following table output:

CATEGORY_ID	BRAND_ID
11	111
12	112
13	113
14	114
15	115
16	116
17	117
18	118
19	119
20	120

Text at the bottom left indicates "10 rows returned in 0.00 seconds". A "CSV Export" link is also present. The bottom right corner shows the application version "Application Express 2.1.0.00.39".

8. customer

```

INSERT INTO customer (customer_id, customer_name, customer_address, customer_phone_no, customer_username, customer_password) VALUES ('masufian', 'Md Abu Sufian', 'Uttara, Dhaka', '01700000011', 'masufian', 'abc');

INSERT INTO customer (customer_id, customer_name, customer_address, customer_phone_no, customer_username, customer_password) VALUES ('smkhan', 'Saidul Mursalin Khan', 'Dhanmondi, Dhaka', '01700000012', 'smkhan', 'abcd');

INSERT INTO customer (customer_id, customer_name, customer_address, customer_phone_no, customer_username, customer_password) VALUES ('masayeed', 'Md Abu Sayeed', 'Nikunja, Dhaka', '01700000013', 'masayeed', 'abcde');

INSERT INTO customer (customer_id, customer_name, customer_address, customer_phone_no, customer_username, customer_password) VALUES ('sksarkar', 'Sumit Kanti Sarkar', 'Kuril, Dhaka', '01700000014', 'sksarkar', 'abcdef');

INSERT INTO customer (customer_id, customer_name, customer_address, customer_phone_no, customer_username, customer_password) VALUES ('aira', 'Abzana Ira', 'Bashundhara, Dhaka', '01700000015', 'aera', 'abcdefg');

INSERT INTO customer (customer_id, customer_name, customer_address, customer_phone_no, customer_username, customer_password) VALUES ('dwarner', 'David Warner', 'Mirpur, Dhaka', '01700000016', 'dwarner', 'abcdefgh');

INSERT INTO customer (customer_id, customer_name, customer_address, customer_phone_no, customer_username, customer_password) VALUES ('hamla', 'Hasim Amla', 'Bashundhara, Dhaka', '01700000017', 'hamla', 'abcdefghi');

INSERT INTO customer (customer_id, customer_name, customer_address, customer_phone_no, customer_username, customer_password) VALUES ('mali', 'Moen Ali', 'Banani, Dhaka', '01700000018', 'mali', 'abcdefgij');

INSERT INTO customer (customer_id, customer_name, customer_address, customer_phone_no, customer_username, customer_password) VALUES ('sali', 'Saif Ali', 'Baridhara, Dhaka', '01700000019', 'sali', 'abcdefgijk');

INSERT INTO customer (customer_id, customer_name, customer_address, customer_phone_no, customer_username, customer_password) VALUES ('niqbal', 'Nafis Iqbal', 'Gulshan, Dhaka', '01700000020', 'niqbal', 'abcdefgijkl');

select * from customer;

```

ORAQUE Database Express Edition

User: ADMIN

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```

SELECT * FROM customer;

```

Results Explain Describe Saved SQL History

CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_ADDRESS	CUSTOMER_PHONE_NO	CUSTOMER_USERNAME	CUSTOMER_PASSWORD
dwarner	David Warner	Mirpur, Dhaka	1700000016	dwarner	abcdefg
hamla	Hasim Amla	Bashundhara, Dhaka	1700000017	hamla	abcdefg
mali	Moen Ali	Banani, Dhaka	1700000018	mali	abcdefgij
sali	Saif Ali	Baridhara, Dhaka	1700000019	sali	abcdefgijk
niqbal	Nafis Iqbal	Gulshan, Dhaka	1700000020	niqbal	abcdefgijkl
masufian	Md Abu Sufian	Uttara, Dhaka	1700000011	masufian	abc
smkhan	Saidul Mursalin Khan	Dhanmondi, Dhaka	1700000012	smkhan	ab
masayeed	Md Abu Sayeed	Nikunja, Dhaka	1700000013	masayeed	abde
sksarkar	Sumit Kanti Sarkar	Kuril, Dhaka	1700000014	sksarkar	abdef

9. wishlist

```
INSERT INTO wishlist (wishlist_id, product_id, customer_id) VALUES (121, 001, 'masufian');

INSERT INTO wishlist (wishlist_id, product_id, customer_id) VALUES (122, 002, 'smkhan');

INSERT INTO wishlist (wishlist_id, product_id, customer_id) VALUES (123, 003, 'masayeed');

INSERT INTO wishlist (wishlist_id, product_id, customer_id) VALUES (124, 004, 'skssarkar');

INSERT INTO wishlist (wishlist_id, product_id, customer_id) VALUES (125, 005, 'aira');

INSERT INTO wishlist (wishlist_id, product_id, customer_id) VALUES (126, 006, 'dwarner');

INSERT INTO wishlist (wishlist_id, product_id, customer_id) VALUES (127, 007, 'hamla');

INSERT INTO wishlist (wishlist_id, product_id, customer_id) VALUES (128, 008, 'mali');

INSERT INTO wishlist (wishlist_id, product_id, customer_id) VALUES (129, 009, 'sali');

INSERT INTO wishlist (wishlist_id, product_id, customer_id) VALUES (130, 010, 'niqbal');

select * from wishlist;
```

The screenshot shows the Oracle Database Express Edition interface. The SQL command window contains the provided SQL code. The results tab shows the data inserted into the wishlist table:

WISHLIST_ID	PRODUCT_ID	CUSTOMER_ID
121	1	masufian
122	2	smkhan
123	3	masayeed
124	4	skssarkar
125	5	aira
126	6	dwarner
127	7	hamla
128	8	mali
129	9	sali
130	10	niqbal

10 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.0.39
Copyright © 1999, 2008, Oracle. All rights reserved.

10. product2

```
INSERT INTO product2 (product_id, product_name, product_type, customer_id) VALUES (001, 'Android Mobile', 'Gadget', 'masufian');

INSERT INTO product2 (product_id, product_name, product_type, customer_id) VALUES (002, 'Core i7 Laptop', 'Gadget', 'smkhan');

INSERT INTO product2 (product_id, product_name, product_type, customer_id) VALUES (003, 'Mens Casual Shirt', 'Fashion', 'masayeed');

INSERT INTO product2 (product_id, product_name, product_type, customer_id) VALUES (004, 'Mens Formal Pant', 'Fashion', 'sksarkar');

INSERT INTO product2 (product_id, product_name, product_type, customer_id) VALUES (005, 'Oracle Database 12c Book', 'Stationary', 'aira');

INSERT INTO product2 (product_id, product_name, product_type, customer_id) VALUES (006, 'Lip Stick', 'Cosmetics', 'dwarner');

INSERT INTO product2 (product_id, product_name, product_type, customer_id) VALUES (007, 'Body Lotion', 'Cosmetics', 'hamla');

INSERT INTO product2 (product_id, product_name, product_type, customer_id) VALUES (008, 'Rice', 'Glossary', 'mali');

INSERT INTO product2 (product_id, product_name, product_type, customer_id) VALUES (009, 'Flour', 'Glossary', 'sali');

INSERT INTO product2 (product_id, product_name, product_type, customer_id) VALUES (010, 'Soyabean Oil', 'Glossary', 'niqbal');

select * from product2
```

The screenshot shows the Oracle Database Express Edition interface. The SQL command window contains the SQL code provided above, with syntax highlighting for keywords and identifiers. The results grid below displays the 10 rows inserted into the product2 table, with columns labeled PRODUCT_ID, PRODUCT_NAME, PRODUCT_TYPE, and CUSTOMER_ID.

PRODUCT_ID	PRODUCT_NAME	PRODUCT_TYPE	CUSTOMER_ID
1	Android Mobile	Gadget	masufian
2	Core i7 Laptop	Gadget	smkhan
3	Mens Casual Shirt	Fashion	masayeed
4	Mens Formal Pant	Fashion	sksarkar
5	Oracle Database 12c Book	Stationary	aira
6	Lip Stick	Cosmetics	dwarner
7	Body Lotion	Cosmetics	hamla
8	Rice	Glossary	mali
9	Flour	Glossary	sali
10	Soyabean Oil	Glossary	niqbal

11. Cart

```
INSERT INTO cart (cart_id, product_id, serial_id, customer_id, pq_id) VALUES (131, 001, 141, 'masufian', 151);
INSERT INTO cart (cart_id, product_id, serial_id, customer_id, pq_id) VALUES (132, 002, 142, 'smkhan', 152);
INSERT INTO cart (cart_id, product_id, serial_id, customer_id, pq_id) VALUES (133, 003, 143, 'masayeed', 153);
INSERT INTO cart (cart_id, product_id, serial_id, customer_id, pq_id) VALUES (134, 004, 144, 'skssarkar', 154);
INSERT INTO cart (cart_id, product_id, serial_id, customer_id, pq_id) VALUES (135, 005, 145, 'aira', 155);
INSERT INTO cart (cart_id, product_id, serial_id, customer_id, pq_id) VALUES (136, 006, 146, 'dawarner', 156);
INSERT INTO cart (cart_id, product_id, serial_id, customer_id, pq_id) VALUES (137, 007, 147, 'hamla', 157);
INSERT INTO cart (cart_id, product_id, serial_id, customer_id, pq_id) VALUES (138, 008, 148, 'mali', 158);
INSERT INTO cart (cart_id, product_id, serial_id, customer_id, pq_id) VALUES (139, 009, 149, 'sali', 159);
INSERT INTO cart (cart_id, product_id, serial_id, customer_id, pq_id) VALUES (140, 010, 150, 'niqbal', 160);
select * from cart;
```

The screenshot shows the Oracle Database Express Edition interface. The SQL command window contains the previously provided SQL script. The results tab is selected, displaying a table with 10 rows of data corresponding to the inserted records. The table has columns: CART_ID, PRODUCT_ID, SERIAL_ID, CUSTOMER_ID, and PQ_ID.

CART_ID	PRODUCT_ID	SERIAL_ID	CUSTOMER_ID	PQ_ID
131	1	141	masufian	151
132	2	142	smkhan	152
133	3	143	masayeed	153
134	4	144	skssarkar	154
135	5	145	aira	155
136	6	146	dawarner	156
137	7	147	hamla	157
138	8	148	mali	158
139	9	149	sali	159
140	10	150	niqbal	160

10 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

12. pqid

```
INSERT INTO pqid (pq_id, price, quantity, total_cost) VALUES (151, 20000, 01, 20000);
INSERT INTO pqid (pq_id, price, quantity, total_cost) VALUES (152, 65000, 01, 65000);
INSERT INTO pqid (pq_id, price, quantity, total_cost) VALUES (153, 1750, 03, 5250);
INSERT INTO pqid (pq_id, price, quantity, total_cost) VALUES (154, 1190, 03, 3570);
INSERT INTO pqid (pq_id, price, quantity, total_cost) VALUES (155, 450, 05, 2250);
INSERT INTO pqid (pq_id, price, quantity, total_cost) VALUES (156, 900, 01, 900);
INSERT INTO pqid (pq_id, price, quantity, total_cost) VALUES (157, 550, 02, 1100);
INSERT INTO pqid (pq_id, price, quantity, total_cost) VALUES (158, 55, 05, 275);
INSERT INTO pqid (pq_id, price, quantity, total_cost) VALUES (159, 46, 02, 92);
INSERT INTO pqid (pq_id, price, quantity, total_cost) VALUES (160, 105, 05, 525);
select * from pqid
```

The screenshot shows the Oracle Database Express Edition interface. The SQL command window contains the previously provided SQL code. The results tab is selected, displaying a table with 10 rows of data:

PQ_ID	PRICE	QUANTITY	TOTAL_COST
151	20000	1	20000
152	65000	1	65000
153	1750	3	5250
154	1190	3	3570
155	450	5	2250
156	900	1	900
157	550	2	1100
158	55	5	275
159	46	2	92
160	105	5	525

10 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

13. payment

```
INSERT INTO payment (payment_id, card, amount, customer_id, cash_on_delivery) VALUES (5551, 21541202, 20000, 'masufian', 'No');

INSERT INTO payment (payment_id, card, amount, customer_id, cash_on_delivery) VALUES (5552, 0, 65000, 'smkhan', 'Yes');

INSERT INTO payment (payment_id, card, amount, customer_id, cash_on_delivery) VALUES (5553, 4154874, 5250, 'masayeed', 'No');

INSERT INTO payment (payment_id, card, amount, customer_id, cash_on_delivery) VALUES (5554, 0, 3570, 'sksarkar', 'Yes');

INSERT INTO payment (payment_id, card, amount, customer_id, cash_on_delivery) VALUES (5555, 0790652, 2250, 'aira', 'No');

INSERT INTO payment (payment_id, card, amount, customer_id, cash_on_delivery) VALUES (5556, 24674520, 900, 'dwarner', 'No');

INSERT INTO payment (payment_id, card, amount, customer_id, cash_on_delivery) VALUES (5557, 21541204, 1100, 'hamla', 'No');

INSERT INTO payment (payment_id, card, amount, customer_id, cash_on_delivery) VALUES (5558, 0, 275, 'mali', 'Yes');

INSERT INTO payment (payment_id, card, amount, customer_id, cash_on_delivery) VALUES (5559, 54251525, 92, 'sali', 'No');

INSERT INTO payment (payment_id, card, amount, customer_id, cash_on_delivery) VALUES (5560, 0, 525, 'niqbal', 'Yes');

select * from payment;
```

ORACLE Database Express Edition

User: ADMIN

Home > SQL > SQL Commands

Autocommit

```
INSERT INTO payment (payment_id, card, amount, customer_id, cash_on_delivery) VALUES (5551, 21541202, 20000, 'masufian', 'No');
INSERT INTO payment (payment_id, card, amount, customer_id, cash_on_delivery) VALUES (5552, 0, 65000, 'smkhan', 'Yes');
INSERT INTO payment (payment_id, card, amount, customer_id, cash_on_delivery) VALUES (5553, 4154874, 5250, 'masayeed', 'No');
INSERT INTO payment (payment_id, card, amount, customer_id, cash_on_delivery) VALUES (5554, 0, 3570, 'sksarkar', 'Yes');
INSERT INTO payment (payment_id, card, amount, customer_id, cash_on_delivery) VALUES (5555, 0790652, 2250, 'aira', 'No');
INSERT INTO payment (payment_id, card, amount, customer_id, cash_on_delivery) VALUES (5556, 24674520, 900, 'dwarner', 'No');
INSERT INTO payment (payment_id, card, amount, customer_id, cash_on_delivery) VALUES (5557, 21541204, 1100, 'hamla', 'No');
INSERT INTO payment (payment_id, card, amount, customer_id, cash_on_delivery) VALUES (5558, 0, 275, 'mali', 'Yes');
INSERT INTO payment (payment_id, card, amount, customer_id, cash_on_delivery) VALUES (5559, 54251525, 92, 'sali', 'No');
INSERT INTO payment (payment_id, card, amount, customer_id, cash_on_delivery) VALUES (5560, 0, 525, 'niqbal', 'Yes');

select * from payment;
```

PAYMENT_ID	CARD	AMOUNT	CUSTOMER_ID	CASH_ON_DELIVERY
5551	21541202	20000	masufian	No
5552	0	65000	smkhan	Yes
5553	4154874	5250	masayeed	No
5554	0	3570	sksarkar	Yes
5555	790652	2250	aira	No
5556	24674520	900	dwarner	No
5557	21541204	1100	hamla	No
5558	0	275	mali	Yes
5559	54251525	92	sali	No
5560	0	525	niqbal	Yes

10 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.0.39
Copyright © 1999, 2008, Oracle. All rights reserved.

14. order1

```
INSERT INTO order1 (order_id, total_cost, product_id, date_of_order, customer_id) VALUES (0101, 20000, 001, DATE '2020-11-22', 'masufian');

INSERT INTO order1 (order_id, total_cost, product_id, date_of_order, customer_id) VALUES (0102, 65000, 002, DATE '2020-11-20', 'smkhan');

INSERT INTO order1 (order_id, total_cost, product_id, date_of_order, customer_id) VALUES (0103, 5250, 003, DATE '2020-11-10', 'masayeed');

INSERT INTO order1 (order_id, total_cost, product_id, date_of_order, customer_id) VALUES (0104, 3570, 004, DATE '2020-11-26', 'sksarkar');

INSERT INTO order1 (order_id, total_cost, product_id, date_of_order, customer_id) VALUES (0105, 2250, 005, DATE '2020-12-12', 'aira');

INSERT INTO order1 (order_id, total_cost, product_id, date_of_order, customer_id) VALUES (0106, 900, 006, DATE '2020-12-11', 'dwarner');

INSERT INTO order1 (order_id, total_cost, product_id, date_of_order, customer_id) VALUES (0107, 1100, 007, DATE '2020-12-11', 'hamla');

INSERT INTO order1 (order_id, total_cost, product_id, date_of_order, customer_id) VALUES (0108, 275, 008, DATE '2020-12-17', 'mali');

INSERT INTO order1 (order_id, total_cost, product_id, date_of_order, customer_id) VALUES (0109, 92, 009, DATE '2020-12-5', 'sali');

INSERT INTO order1 (order_id, total_cost, product_id, date_of_order, customer_id) VALUES (0110, 525, 010, DATE '2020-12-4', 'niqbal');

select * from order1;
```

ORACLE Database Express Edition

User: ADMIN

Home > SQL > SQL Commands

Autocommit

```
INSERT INTO order1 (order_id, total_cost, product_id, date_of_order, customer_id) VALUES (0101, 20000, 001, DATE '2020-11-22', 'masufian');
INSERT INTO order1 (order_id, total_cost, product_id, date_of_order, customer_id) VALUES (0102, 65000, 002, DATE '2020-11-20', 'smkhan');
INSERT INTO order1 (order_id, total_cost, product_id, date_of_order, customer_id) VALUES (0103, 5250, 003, DATE '2020-11-10', 'masayeed');
INSERT INTO order1 (order_id, total_cost, product_id, date_of_order, customer_id) VALUES (0104, 3570, 004, DATE '2020-11-26', 'sksarkar');
INSERT INTO order1 (order_id, total_cost, product_id, date_of_order, customer_id) VALUES (0105, 2250, 005, DATE '2020-12-12', 'aira');
INSERT INTO order1 (order_id, total_cost, product_id, date_of_order, customer_id) VALUES (0106, 900, 006, DATE '2020-12-11', 'dwarner');
INSERT INTO order1 (order_id, total_cost, product_id, date_of_order, customer_id) VALUES (0107, 1100, 007, DATE '2020-12-11', 'hamla');
INSERT INTO order1 (order_id, total_cost, product_id, date_of_order, customer_id) VALUES (0108, 275, 008, DATE '2020-12-17', 'mali');
INSERT INTO order1 (order_id, total_cost, product_id, date_of_order, customer_id) VALUES (0109, 92, 009, DATE '2020-12-5', 'sali');
INSERT INTO order1 (order_id, total_cost, product_id, date_of_order, customer_id) VALUES (0110, 525, 010, DATE '2020-12-4', 'niqbal');

select * from order1;
```

Results Explain Describe Saved SQL History

ORDER_ID	TOTAL_COST	PRODUCT_ID	DATE_OF_ORDER	CUSTOMER_ID
101	20000	1	22-NOV-20	masufian
102	65000	2	20-NOV-20	smkhan
103	5250	3	10-NOV-20	masayeed
104	3570	4	28-NOV-20	sksarkar
105	2250	5	12-DEC-20	aira
106	900	6	11-DEC-20	dwarner
107	1100	7	11-DEC-20	hamla
108	275	8	17-DEC-20	mali
109	92	9	05-DEC-20	sali
110	525	10	04-DEC-20	niqbal

10 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2008, Oracle. All rights reserved.

15. product3

```
INSERT INTO product3 (product_id, product_name, product_type, cart_id) VALUES (001, 'Android Mobile', 'Gadget', 131);

INSERT INTO product3 (product_id, product_name, product_type, cart_id) VALUES (002, 'Core i7 Laptop', 'Gadget', 132);

INSERT INTO product3 (product_id, product_name, product_type, cart_id) VALUES (003, 'Mens Casual Shirt', 'Fashion', 133);

INSERT INTO product3 (product_id, product_name, product_type, cart_id) VALUES (004, 'Mens Formal Pant', 'Fashion', 134);

INSERT INTO product3 (product_id, product_name, product_type, cart_id) VALUES (005, 'Oracle Database 12c Book', 'Stationary', 135);

INSERT INTO product3 (product_id, product_name, product_type, cart_id) VALUES (006, 'Lip Stick', 'Cosmetics', 136);

INSERT INTO product3 (product_id, product_name, product_type, cart_id) VALUES (007, 'Body Lotion', 'Cosmetics', 137);

INSERT INTO product3 (product_id, product_name, product_type, cart_id) VALUES (008, 'Rice', 'Glossary', 138);

INSERT INTO product3 (product_id, product_name, product_type, cart_id) VALUES (009, 'Flour', 'Glossary', 139);

INSERT INTO product3 (product_id, product_name, product_type, cart_id) VALUES (010, 'Soyabean Oil', 'Glossary', 140);

select * from product3;
```

The screenshot shows the Oracle Database Express Edition interface. The SQL command window contains the previously provided SQL script. The results tab shows a table with 10 rows of data, which matches the expected output for the product3 table.

PRODUCT_ID	PRODUCT_NAME	PRODUCT_TYPE	CART_ID
1	Android Mobile	Gadget	131
2	Core i7 Laptop	Gadget	132
3	Mens Casual Shirt	Fashion	133
4	Mens Formal Pant	Fashion	134
5	Oracle Database 12c Book	Stationary	135
6	Lip Stick	Cosmetics	136
7	Body Lotion	Cosmetics	137
8	Rice	Glossary	138
9	Flour	Glossary	139
10	Soyabean Oil	Glossary	140

16. order2

```
INSERT INTO order2 (order_id, total_cost, product_id, date_of_order, cart_id) VALUES (0101, 20000, 001, DATE '2020-11-22', 131);

INSERT INTO order2 (order_id, total_cost, product_id, date_of_order, cart_id) VALUES (0102, 65000, 002, DATE '2020-11-20', 132);

INSERT INTO order2 (order_id, total_cost, product_id, date_of_order, cart_id) VALUES (0103, 5250, 003, DATE '2020-11-10', 133);

INSERT INTO order2 (order_id, total_cost, product_id, date_of_order, cart_id) VALUES (0104, 3570, 004, DATE '2020-11-26', 134);

INSERT INTO order2 (order_id, total_cost, product_id, date_of_order, cart_id) VALUES (0105, 2250, 005, DATE '2020-12-12', 135);

INSERT INTO order2 (order_id, total_cost, product_id, date_of_order, cart_id) VALUES (0106, 900, 006, DATE '2020-12-11', 136);

INSERT INTO order2 (order_id, total_cost, product_id, date_of_order, cart_id) VALUES (0107, 1100, 007, DATE '2020-12-11', 137);

INSERT INTO order2 (order_id, total_cost, product_id, date_of_order, cart_id) VALUES (0108, 275, 008, DATE '2020-12-17', 138);

INSERT INTO order2 (order_id, total_cost, product_id, date_of_order, cart_id) VALUES (0109, 92, 009, DATE '2020-12-5', 139);

INSERT INTO order2 (order_id, total_cost, product_id, date_of_order, cart_id) VALUES (0110, 525, 010, DATE '2020-12-4', 140);

select * from order2;
```

The screenshot shows the Oracle Database Express Edition interface. The SQL command window displays the executed SQL statements and their results. The results section shows a table with 10 rows of data.

ORDER_ID	TOTAL_COST	PRODUCT_ID	DATE_OF_ORDER	CART_ID
101	20000	1	22-NOV-20	131
102	65000	2	20-NOV-20	132
103	5250	3	10-NOV-20	133
104	3570	4	26-NOV-20	134
105	2250	5	12-DEC-20	135
106	900	6	11-DEC-20	136
107	1100	7	11-DEC-20	137
108	275	8	17-DEC-20	138
109	92	9	05-DEC-20	139
110	525	10	04-DEC-20	140

17. payment2

```
INSERT INTO payment2 (payment_id, card, amount, order_id, cash_on_delivery) VALUES (5551, 21541202, 20000, 0101, 'No');

INSERT INTO payment2 (payment_id, card, amount, order_id, cash_on_delivery) VALUES (5552, 0, 65000, 0102, 'Yes');

INSERT INTO payment2 (payment_id, card, amount, order_id, cash_on_delivery) VALUES (5553, 4154874, 5250, 0103, 'No');

INSERT INTO payment2 (payment_id, card, amount, order_id, cash_on_delivery) VALUES (5554, 0, 3570, 0104, 'Yes');

INSERT INTO payment2 (payment_id, card, amount, order_id, cash_on_delivery) VALUES (5555, 0790652, 2250, 0105, 'No');

INSERT INTO payment2 (payment_id, card, amount, order_id, cash_on_delivery) VALUES (5556, 24674520, 900, 0106, 'No');

INSERT INTO payment2 (payment_id, card, amount, order_id, cash_on_delivery) VALUES (5557, 21541204, 1100, 0107, 'No');

INSERT INTO payment2 (payment_id, card, amount, order_id, cash_on_delivery) VALUES (5558, 0, 275, 0108, 'Yes');

INSERT INTO payment2 (payment_id, card, amount, order_id, cash_on_delivery) VALUES (5559, 54251525, 92, 0109, 'No');

INSERT INTO payment2 (payment_id, card, amount, order_id, cash_on_delivery) VALUES (5560, 0, 525, 0110, 'Yes');

select * from payment2
```

The screenshot shows the Oracle Database Express Edition interface. The SQL command window displays the previously written SQL statements for inserting data into the payment2 table. The results tab shows the data inserted, which is identical to the table shown in the screenshot.

PAYMENT_ID	CARD	AMOUNT	ORDER_ID	CASH_ON_DELIVERY
5551	21541202	20000	101	No
5552	0	65000	102	Yes
5553	4154874	5250	103	No
5554	0	3570	104	Yes
5555	0790652	2250	105	No
5556	24674520	900	106	No
5557	21541204	1100	107	No
5558	0	275	108	Yes
5559	54251525	92	109	No
5560	0	525	110	Yes

Query Writing

1. Sub Query

1-Q:

Print the product id, product names, product type from product table where product_id is 1.

1-A:

```
select product_id, product_name, product_type from product where product_name = (select product_name from product2 where product_id=1);
```

The screenshot shows the Oracle Database Express Edition interface. The SQL command window contains the following code:

```
select product_id, product_name, product_type from product where product_name = (select product_name from product2 where product_id=1)
```

The results pane displays a single row of data:

PRODUCT_ID	PRODUCT_NAME	PRODUCT_TYPE
1	Android Mobile	Gadget

Below the results, it says "1 rows returned in 0.00 seconds". The bottom status bar indicates "Language: en-us" and "Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved."

2-Q:

Print product id, product name, product type and cart id for cart id 131.

2-A:

```
SELECT product_id, product_name, product_type FROM product3 WHERE cart_id = (SELECT cart_id FROM cart WHERE cart_id ='131');
```

ORACLE Database Express Edition

User: ADMIN

Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT product_id, product_name, product_type FROM product3 WHERE cart_id = (SELECT cart_id FROM cart WHERE cart_id ='135');
```

Results Explain Describe Saved SQL History

PRODUCT_ID	PRODUCT_NAME	PRODUCT_TYPE
5	Oracle Database 12c Book	Stationary

1 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.00.39
Copyright © 1999, 2008, Oracle. All rights reserved.

3-Q:

Print order id, total cost, product id, date of order, cart_id from order2 table where the total cost is greater than the cost of order which were placed on 2020-11-26.

3-A:

```
SELECT order_id, total_cost, product_id, date_of_order, cart_id FROM order2 WHERE  
total_cost > (SELECT AVG(total_cost) FROM order2 WHERE date_of_order=to_date('2020-11-  
26', 'YYYY-MM-DD'));
```

The screenshot shows the Oracle Database Express Edition interface. At the top, it says "ORACLE Database Express Edition" and "User: ADMIN". Below that is a toolbar with icons for Home, Logout, and Help. The main area is titled "Home > SQL > SQL Commands". A dropdown menu shows "Autocommit" is checked and "Display" is set to 10. There are "Save" and "Run" buttons. The SQL command entered is:

```
SELECT order_id, total_cost, product_id, date_of_order, cart_id FROM order2 WHERE total_cost > (SELECT AVG(total_cost) FROM order2 WHERE date_of_order=to_date('2020-11-26', 'YYYY-MM-DD'));
```

Below the command is a large empty box for results. Underneath is a navigation bar with "Results", "Explain", "Describe", "Saved SQL", and "History". The results section shows a table with the following data:

ORDER_ID	TOTAL_COST	PRODUCT_ID	DATE_OF_ORDER	CART_ID
101	20000	1	22-NOV-20	131
102	65000	2	20-NOV-20	132
103	5250	3	10-NOV-20	133

Below the table, it says "3 rows returned in 0.00 seconds" and there is a "CSV Export" link. At the bottom, it says "Language: en-us" and "Application Express 2.1.0.00.39 Copyright © 1999, 2008, Oracle. All rights reserved."

2. Joining

1-Q:

Print the admin name and admin address id by joining admin table and address table

1-A:

```
SELECT a.admin_name "Admin Name", ad.address_id "Admin Address ID" FROM admin a,  
address ad WHERE a.address_id = ad.address_id;
```

The screenshot shows the Oracle Database Express Edition interface. The SQL command window contains the following code:

```
SELECT a.admin_name "Admin Name", ad.address_id "Admin Address" FROM admin a, address ad WHERE a.address_id = ad.address_id;
```

The results section displays a single row of data:

Admin Name	Admin Address
alex	Dhaka,Bangladesh

1 rows returned in 0.00 seconds [CSV Export](#)

At the bottom, it shows the language is en-us and the application version is Application Express 2.1.0.00.39. Copyright © 1999, 2006, Oracle. All rights reserved.

2-Q:

Print the customers name for customer id masufian, smkhan and aira

2-A:

```
SELECT cname.customer_name FROM customer cname, customer cid WHERE  
cname.customer_id = cid.customer_id AND cname.customer_id in('masufian', 'smkhan', 'aira');
```

ORACLE Database Express Edition

User: ADMIN

Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT cname.customer_name FROM customer cname, customer cid WHERE cname.customer_id = cid.customer_id AND cname.customer_id in('masufian', 'smkhan',  
'aira');
```

Results Explain Describe Saved SQL History

CUSTOMER_NAME
Abzana Ira
Md Abu Sufian
Saidul Mursalin Khan

3 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.00.39
Copyright © 1999, 2008, Oracle. All rights reserved.

3-Q:

Print the customer id and count the order id from order1 table and join the customer table with order1 table.

3-A:

```
SELECT customer_id, COUNT( order_id ) FROM order1 INNER JOIN customer  
USING(customer_id) GROUP BY customer_id ORDER BY customer_id;
```

The screenshot shows the Oracle Database Express Edition interface. The SQL command window contains the following SQL query:

```
SELECT customer_id, COUNT( order_id ) FROM order1 INNER JOIN customer USING(customer_id) GROUP BY customer_id ORDER BY customer_id;
```

The results section displays a table with two columns: CUSTOMER_ID and COUNT(ORDER_ID). The data is as follows:

CUSTOMER_ID	COUNT(ORDER_ID)
aira	1
dwarner	1
hamla	1
mali	1
masayeed	1
masufian	1
niqbal	1
sali	1
skarkar	1
smkhan	1

Below the table, it says "10 rows returned in 0.00 seconds".

Application Express 2.1.0.00.39

3. Group Function

1-Q:

Find average, maximum, minimum salary of the pqid.

1-A:

select avg(price) as average,max(price) as maximum,min(price) as minimum from pqid;

The screenshot shows the Oracle Database Express Edition interface. The SQL command window contains the following code:

```
select avg(price) as average,max(price) as maximum,min(price) as minimum from pqid;
```

The results section displays the following table:

AVERAGE	MAXIMUM	MINIMUM
9004.6	65000	46

Below the table, it says "1 rows returned in 0.00 seconds".

At the bottom, there are links for "Results", "Explain", "Describe", "Saved SQL", and "History". The status bar at the bottom right shows "Application Express 2.1.0.00.39" and "Copyright © 1999, 2006, Oracle. All rights reserved."

2-Q:

Display the maximum amount of payment2

2-A:

select max(amount) from payment2;

The screenshot shows the Oracle Database Express Edition interface. The SQL command entered is:

```
select max(amount) from payment2
```

The results section displays a single row:

MAX(AMOUNT)
65000

Below the results, it says "1 rows returned in 0.00 seconds".

At the bottom, there are links for "Language: en-gb", "Application Express 2.1.0.00.39", and "Copyright © 1999, 2006, Oracle. All rights reserved."

3-Q:

Find the number of customer who have payment <3570

3-A:

select count(customer_id) from payment where amount<3570;

The screenshot shows the Oracle Database Express Edition interface. The SQL command entered is:

```
select count(customer_id) from payment where amount<3570
```

The results section displays:

COUNT(CUSTOMER_ID)
6

1 rows returned in 0.00 seconds

CSV Export

Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.

4. Single row function

1-Q:

print product, total price from order2 table and print the total cost in ascending order

1-A:

```
select product_id "Product ID", TRUNC(total_cost) "Total Price" from order2 order by total_cost asc
```

The screenshot shows the Oracle Database Express Edition interface. The SQL command window contains the following SQL query:

```
select product_id "Product ID", TRUNC(total_cost) "Total Price" from order2 order by total_cost asc;
```

The results section displays a table with two columns: Product ID and Total Price. The data is as follows:

Product ID	Total Price
9	92
8	275
10	525
6	900
7	1100
5	2250
4	3570
3	5250
1	20000
2	65000

Below the table, it says "10 rows returned in 0.00 seconds".

5. View

1-Q:

Create a view for customer table where we are going to print Customer ID, Customer Name, Phone Number and Address from customer table.

1-A:

```
CREATE VIEW customer_view AS
```

```
SELECT c.customer_id "Customer ID", c.customer_name "Customer Name",
c.customer_phone_no "Phone Number", c.customer_address "Address" FROM customer c;
```

The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window contains the following code:

```
CREATE VIEW customer_view AS
SELECT c.customer_id "Customer ID", c.customer_name "Customer Name", c.customer_phone_no "Phone Number", c.customer_address "Address" FROM customer c;
select * from customer_view
```

The results window displays the data from the customer table:

Customer ID	Customer Name	Phone Number	Address
dwarmer	David Warner	1700000016	Mirpur,Dhaka
hamla	Hasim Amla	1700000017	Bashundhara,Dhaka
mali	Moeni Ali	1700000018	Banani,Dhaka
sali	Saif Ali	1700000019	Baridhara,Dhaka
niqbal	Nafis Iqbal	1700000020	Gulshan,Dhaka
masufian	Md Abu Sufian	1700000011	Uttara, Dhaka
smkhan	Saidul Mursalin Khan	1700000012	Dhanmondi, Dhaka
masayeed	Md Abu Sayeed	1700000013	Nikunja, Dhaka
sksarkar	Sumit Kanti Sarkar	1700000014	Kuril, Dhaka
aira	Abzana Ira	1700000015	Bashundhara,Dhaka

10 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2008, Oracle. All rights reserved.

Sequence

➤ **Table(shipping) Create for sequence:**

```
CREATE TABLE shipping
(
    shipping_id NUMBER(8) NOT NULL,
    customer_id VARCHAR2(20) NOT NULL,
    date_of_order DATE,
    order_id NUMBER(4) NOT NULL,
    product_id NUMBER(4) NOT NULL,
    payment_id NUMBER(4) NOT NULL,
    customer_address VARCHAR2(50) NOT NULL,
    customer_phone_no NUMBER(11) NOT NULL,
    CONSTRAINT shipping_PK PRIMARY KEY(shipping_id),
    CONSTRAINT shipping_customer_FK FOREIGN KEY(customer_id) REFERENCES customer(customer_id),
    CONSTRAINT shipping_order_FK FOREIGN KEY(order_id) REFERENCES order1(order_id),
    CONSTRAINT shipping_product_FK FOREIGN KEY(product_id) REFERENCES product(product_id),
    CONSTRAINT shipping_payment_FK FOREIGN KEY(payment_id) REFERENCES payment(payment_id),
    CONSTRAINT shipping_unique UNIQUE (shipping_id, customer_id, order_id, payment_id, product_id)
);
```

➤ **Create Sequence:**

```
CREATE SEQUENCE shipping_id  
START WITH 1230  
INCREMENT BY 1  
MINVALUE 1230  
MAXVALUE 10000  
NOCACHE  
NOCYCLE;  
  
CREATE SEQUENCE order_id  
START WITH 100  
INCREMENT BY 1  
MINVALUE 100  
MAXVALUE 10000  
NOCACHE  
NOCYCLE;  
  
CREATE SEQUENCE product_id  
START WITH 0  
INCREMENT BY 1  
MINVALUE 0  
MAXVALUE 10000  
NOCACHE  
NOCYCLE;  
  
CREATE SEQUENCE payment_id  
START WITH 5550  
INCREMENT BY 1  
MINVALUE 5550  
MAXVALUE 10000  
NOCACHE  
NOCYCLE;
```

➤ **Data Insertion in shipping table:**

```
INSERT INTO shipping (shipping_id, customer_id, order_id, product_id, payment_id, date_of_order,
customer_phone_no, customer_address) VALUES (shipping_id.NEXTVAL, 'masufian', order_id.NEXTVAL,
product_id.NEXTVAL, payment_id.NEXTVAL, DATE '2020-11-22', 01700000011, 'Uttara, Dhaka');

INSERT INTO shipping (shipping_id, customer_id, order_id, product_id, payment_id, date_of_order,
customer_phone_no, customer_address) VALUES (shipping_id.NEXTVAL, 'smkhan', order_id.NEXTVAL,
product_id.NEXTVAL, payment_id.NEXTVAL, DATE '2020-11-20', 01700000012, 'Dhanmondi, Dhaka');

INSERT INTO shipping (shipping_id, customer_id, order_id, product_id, payment_id, date_of_order,
customer_phone_no, customer_address) VALUES (shipping_id.NEXTVAL, 'masayeed', order_id.NEXTVAL,
product_id.NEXTVAL, payment_id.NEXTVAL, DATE '2020-11-10', 01700000013, 'Nikunja, Dhaka');

INSERT INTO shipping (shipping_id, customer_id, order_id, product_id, payment_id, date_of_order,
customer_phone_no, customer_address) VALUES (shipping_id.NEXTVAL, 'sksarkar', order_id.NEXTVAL,
product_id.NEXTVAL, payment_id.NEXTVAL, DATE '2020-11-26', 01700000014, 'Kuril, Dhaka');

INSERT INTO shipping (shipping_id, customer_id, order_id, product_id, payment_id, date_of_order,
customer_phone_no, customer_address) VALUES (shipping_id.NEXTVAL, 'aira', order_id.NEXTVAL, product_id.NEXTVAL,
payment_id.NEXTVAL, DATE '2020-12-12', 01700000015, 'Bashundhara, Dhaka');

INSERT INTO shipping (shipping_id, customer_id, order_id, product_id, payment_id, date_of_order,
customer_phone_no, customer_address) VALUES (shipping_id.NEXTVAL, 'dwarner', order_id.NEXTVAL,
product_id.NEXTVAL, payment_id.NEXTVAL, DATE '2020-12-11', 01700000016, 'Mirpur,Dhaka');

INSERT INTO shipping (shipping_id, customer_id, order_id, product_id, payment_id, date_of_order,
customer_phone_no, customer_address) VALUES (shipping_id.NEXTVAL, 'hamla', order_id.NEXTVAL,
product_id.NEXTVAL, payment_id.NEXTVAL, DATE '2020-12-11', 01700000017, 'Bashundhara,Dhaka');

INSERT INTO shipping (shipping_id, customer_id, order_id, product_id, payment_id, date_of_order,
customer_phone_no, customer_address) VALUES (shipping_id.NEXTVAL, 'mali', order_id.NEXTVAL, product_id.NEXTVAL,
payment_id.NEXTVAL, DATE '2020-12-17', 01700000018, 'Banani,Dhaka');

INSERT INTO shipping (shipping_id, customer_id, order_id, product_id, payment_id, date_of_order,
customer_phone_no, customer_address) VALUES (shipping_id.NEXTVAL, 'sali', order_id.NEXTVAL, product_id.NEXTVAL,
payment_id.NEXTVAL, DATE '2020-12-5', 01700000019, 'Baridhara,Dhaka');

INSERT INTO shipping (shipping_id, customer_id, order_id, product_id, payment_id, date_of_order,
customer_phone_no, customer_address) VALUES (shipping_id.NEXTVAL, 'niqbal', order_id.NEXTVAL,
product_id.NEXTVAL, payment_id.NEXTVAL, DATE '2020-12-4', 01700000020, 'Gulshan,Dhaka');

select * from shipping
```

User: ADMIN

Home > SQL > SQL Commands

Autocommit

```
INSERT INTO shipping (shipping_id, customer_id, order_id, product_id, payment_id, date_of_order, customer_phone_no, customer_address) VALUES (shipping_id.NEXTVAL, 'mali', order_id.NEXTVAL, product_id.NEXTVAL, payment_id.NEXTVAL, DATE '2020-12-17', 01700000018, 'Banani,Dhaka');

INSERT INTO shipping (shipping_id, customer_id, order_id, product_id, payment_id, date_of_order, customer_phone_no, customer_address) VALUES (shipping_id.NEXTVAL, 'sali', order_id.NEXTVAL, product_id.NEXTVAL, payment_id.NEXTVAL, DATE '2020-12-5', 01700000019, 'Baridhara,Dhaka');

INSERT INTO shipping (shipping_id, customer_id, order_id, product_id, payment_id, date_of_order, customer_phone_no, customer_address) VALUES (shipping_id.NEXTVAL, 'niqbal', order_id.NEXTVAL, product_id.NEXTVAL, payment_id.NEXTVAL, DATE '2020-12-4', 01700000020, 'Gulshan,Dhaka');
```

`Select * from shipping`

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

SHIPPING_ID	CUSTOMER_ID	DATE_OF_ORDER	ORDER_ID	PRODUCT_ID	PAYMENT_ID	CUSTOMER_ADDRESS	CUSTOMER_PHONE_NO
1231	masufian	22-NOV-20	101	1	5561	Uttara, Dhaka	1700000011
1232	smkhan	20-NOV-20	102	2	5562	Dhanmondi, Dhaka	1700000012
1233	masayeed	10-NOV-20	103	3	5563	Nikunja, Dhaka	1700000013
1234	sksarkar	28-NOV-20	104	4	5564	Kuril, Dhaka	1700000014
1235	aira	12-DEC-20	105	5	5565	Bashundhara, Dhaka	1700000015
1236	dwarmer	11-DEC-20	106	6	5566	Mirpur,Dhaka	1700000016
1237	hamla	11-DEC-20	107	7	5567	Bashundhara,Dhaka	1700000017
1238	mali	17-DEC-20	108	8	5568	Banani,Dhaka	1700000018
1239	sali	05-DEC-20	109	9	5569	Baridhara,Dhaka	1700000019
1240	niqbal	04-DEC-20	110	10	5570	Gulshan,Dhaka	1700000020

10 rows returned in 0.00 seconds

[CSV Export](#)

Application Express 2.1.0.00.39

Copyright © 1999, 2008, Oracle. All rights reserved.

Language: en-us

Relational Algebra

1. $\prod \text{customer}_{\text{name}} (\sigma_{\text{customer_id}=\text{"smkhan"}} \text{ (customer)})$
2. $\prod \text{product}_{\text{name}}, \text{product}_{\text{type}} (\sigma_{\text{product_id}=5} \text{ (product)})$
3. $\prod \text{category}_{\text{name}} (\sigma_{\text{category_id}=\text{"15}}} \text{ (category)})$
4. $\prod \text{a.admin}_{\text{name}}, \text{add.admin}_{\text{address}} (\sigma_{\text{a.address_id} = \text{add.address_id} \text{ and } \text{a.address_id} = 1} \text{ (admin } a \times \text{address add)})$
5. $\prod \text{order}_{\text{id}} (\sigma_{\text{date_of_order}=\text{"2020-12-12}} \text{ (order1)})$

Conclusion

Finally, successfully develop and implement the E-commerce Management System with the help of Oracle Database 10g and SQL query, includes all the features which are the fundamental requirements for an E-commerce Management System.

Finally got success in our attempt to take care of the needs of both the customers as well as the administrator which was our main objective.