

AMERICAN INTERNATIONAL UNIVERSITY BANGLADESH
(AIUB)

FACULTY OF SCIENCE & TECHNOLOGY



Course Title
INTRODUCTION TO DATABASE (2108)

Semester: 2023-2024, Spring

Section: [D]

TITLE
Gadget and Gear Management System

Supervised By
Sifat Rahman Ahona

Submitted By: Group no: 01

Name	ID
Afnan Bin Islam Nahin	22-49350-3
Joy Sarker	22-49903-3
Aman Ullah	22-49345-3
S M Sarar Sakib	22-49900-3

TABLE OF CONTENTS

TOPICS	Page no.
Title Page	1
Table of Content	2
1. Introduction	3
2. Case Study	4
3. ER Diagram	5
4. Normalization	6-8
5. Finalization	9
6. Table Creation	10-13
7. Data Insertion	14-17
8. Query Test	18-27
9. Relational Algebra	28
10. Conclusion	29

Introduction

Welcome to the Gadget and Gear Management System (GGMS) database project. GGMS is crafted to address the contemporary challenges in managing electronic devices within organizations. This project aims to develop a centralized platform for efficient inventory tracking, maintenance scheduling, and allocation of gadgets and gears. By providing real-time visibility and optimization of resources, GGMS intends to enhance productivity and reduce operational costs. Through this project, we endeavor to design a user-friendly interface with robust functionality, revolutionizing the management of gadgets and gears within organizations.

Case Study / Scenario

StudentID1: 22-49350-3 Name: Afnan Bin Islam Nahin	StudentID3: 22-49903-3 Name: Joy Sarker
StudentID2: 22-49345-3 Name: Aman Ullah	StudentID4: 22-49900-3 Name: S M Sarar Sakib
CO2: Understand the fundamental concepts underlying database systems and gain hands-on experience with ER diagram Case study	
PO-c2: Develop process for complex computer science and engineering problems considering cultural and societal factors.	Marks

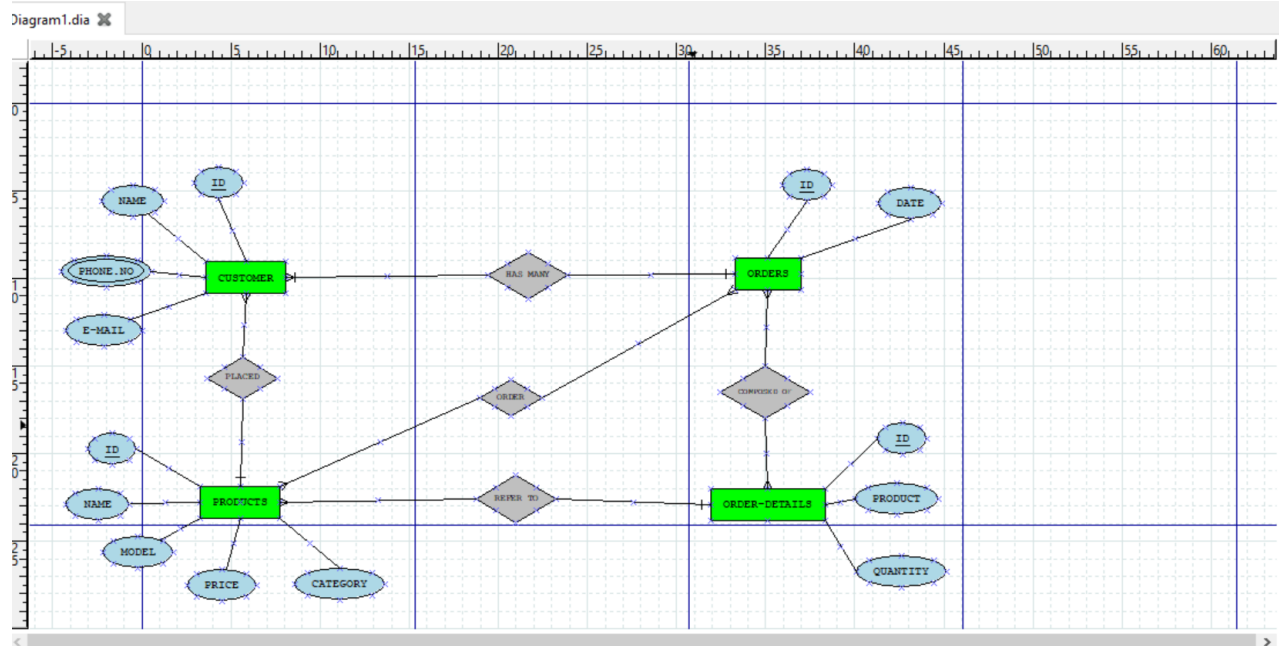
Gadget and Gear Management System (GGMS) is enhancing operational efficiency with a design that allows customers to order a variety of tech products based on their preferences. It's an online-based company.

The company offers a wide range of electronics, including Apple mobile phones (iPhone 13 Pro to iPhone 15 Pro), Apple laptops (Macbook), headphones, shooting drones, power banks, speakers, camaras, and smart watches. Additionally, TVs from Sony are available. Customers can order products and must provide their name, phone number, and email.

This order details the product name, model, quantity, price, and order date. Orders are managed centrally, streamlining the purchasing process for efficiency and customer satisfaction. This company sells products at lower prices than others. So the sales of every product from this company have increased. Struggling to handle customers. Many customers buy more than one product due to lower prices. Due to this, there are many customers who are not able to buy any products.

For this reason, the company made a policy which is each customer can purchase any one product. These policy are made so that the products of this company can reach every customer.

ER Diagram



Normalization

Placed (C.Id, C.Name, Phone No., Email, P.Id, P.Name, Model, Price, Category)

1NF: Phone No. Multivalued Attributes

2NF: C.Id, C.Name, Phone No., Email, P.Id

P.Id, P.Name, Model, Price, Category

3NF: C.Id, C.Name, Phone No., Email, P.Id

P.Id, P.Name, Model, Price, Category.

(No Transitive Dependency)

Table:

C.Id, C.Name, Phone No., Email, P.Id

P.Id, P.Name, Model, Price, Category.

Has Many (C.Id, C.Name, Phone No., Email, O.Id, Date)

1NF: Phone No. Multivalued Attributes

2NF: C.Id, C.Name, Phone No., Email, O.Id

O.Id, Date

3NF: C.Id, C.Name, Phone No, Email, O.Id

O.Id, Date

(No Transitive Dependency)

Table:

C.Id, C.Name, Phone No., Email, O.Id

O.Id, Date

Order (P.Id, P.Name, Model, Price, Category, O.Id, Date)

1NF: No Multivalued Attributes

2NF: P.Id, P.Name, Model, Price, Category

O.Id, Date

PO.Id, P.Id, O.Id

3NF: P.Id, P.Name, Model, Price, Category

O.Id, Date

PO.Id, P.Id, O.Id

(No Transitive Dependency)

Table:

P.Id, P.Name, Model, Price, Category

O.Id, Date.

PO.Id, P.Id, O.Id

Refer To (P.Id, P.Name, Model, Price, Category, Od.Id, Quantity, Product-Id)

1NF: No Multivalued Attributes

2NF: P.Id, P.Name, Model, Price, Category, Od.Id

Od.Id, Quantity, Product-Id

3NF: P.Id, P.Name, Model, Price, Category, Od.Id

Od.Id, Quantity, Product

Table:

P.Id, P.Name, Model, Price, Category, Od.Id

Od.Id, Quantity, Product

Composed Of (O.Id, Date, Od.Id, Product, Quantity)

1NF: No Multivalued Attributes

2NF: O.Id, Date

Od.Id, Quantity, Product

Ood.Id, O.Id, Od.Id

3NF: O.Id, Date

Od.Id, Quantity, Product

Ood.Id, O.Id, Od.Id

Table:

O.Id, Date

Od.Id, Quantity, Product

Ood.Id, O.Id, Od.Id

Total Table:

1)C.Id, C.Name, Phone No., Email, P.Id

2)P.Id, P.Name, Model, Price, Category

3)C.Id, C.Name, Phone No., Email, O.Id

4)O.Id, Date

5)P.Id, P.Name, Model, Price, Category X

6)O.Id, Date. X

7)PO.Id, P.Id, O.Id

8)P.Id, P.Name, Model, Price, Category, Od.Id

9)Od.Id, Quantity, Product

10)O.Id, Date X

11)Od.Id, Quantity, Product X

12)Ood.Id, O.Id, Od.Id--Delivery

Finalization

Final Table:

- 1) C.Id, C.Name, Phone No., Email, P.Id--**Customer Details**
- 2) P.Id, P.Name, Model, Price, Category—**Product Information**
- 3) C.Id, C.Name, Phone No., Email, O.Id—**Customer Information**
- 4) O.Id, Date—**Order Details**
- 5) Po.Id, P.Id, O.Id—**Order Details 1**
- 6) P.Id, P.Name, Model, Price, Category, Od.Id—**Product Details**
- 7) Od.Id, Quantity, Product—**Product Details 1**
- 8) Ood.Id, O.Id, Od.Id--**Delivery**

Table Creation (DDL Operations)

StudentID1: 22-49350-3 Name: Afnan Bin Islam Nahin	StudentID3: 22-49903-3 Name: Joy Sarker
StudentID2: 22-49345-3 Name: Aman Ullah	StudentID4: 22-49900-3 Name: S M Sarar Sakib
CO4: Creating DML, DDL using Oracle and connection with ODBC/JDBC for existing JAVA application	
PO-e-2: Use modern engineering and IT tools for prediction and modeling of complex computer science and engineering problem	Marks

Create Table : Customer Details

The screenshot shows the Oracle SQL Developer interface. The SQL Command window contains the following SQL code:

```

CREATE TABLE CUSTOMER_DETAILS (
  CUSTOMER_ID NUMBER(10,0) PRIMARY KEY,
  CUSTOMER_NAME VARCHAR2(50),
  PHONE_NUMBER VARCHAR2(50),
  PRODUCT_ID NUMBER(10,0),
  EMAIL VARCHAR2(50)
);
DESC CUSTOMER_DETAILS
ALTER TABLE CUSTOMER_DETAILS ADD CONSTRAINT PLACED_fk FOREIGN KEY (PRODUCT_ID) REFERENCES PRODUCT_INFORMATION(PRODUCT_ID);
  
```

Below the SQL window, the 'Results' tab displays the table structure for 'CUSTOMER_DETAILS':

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER_DETAILS	CUSTOMER_ID	Number	-	10	0	1	-	-	-
	CUSTOMER_NAME	Varchar2	50	-	-	-	✓	-	-
	PHONE_NUMBER	Varchar2	50	-	-	-	✓	-	-
	PRODUCT_ID	Number	-	10	0	-	✓	-	-
	EMAIL	Varchar2	50	-	-	-	✓	-	-

The footer of the application shows 'Application Express 2.1.0.0.39' and 'Copyright © 1999, 2006, Oracle. All rights reserved.'

Create Table : Product Information

The screenshot shows the Oracle SQL Developer interface. The SQL Command window contains the following SQL code:

```

CREATE TABLE PRODUCT_INFORMATION (
  PRODUCT_ID NUMBER(10,0) PRIMARY KEY,
  PRODUCT_NAME VARCHAR2(50),
  MODEL VARCHAR2(50),
  PRICE NUMBER(10,0),
  CATEGORY VARCHAR2(50)
);
DESC PRODUCT_INFORMATION
  
```

Below the SQL window, the 'Results' tab displays the table structure for 'PRODUCT_INFORMATION':

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PRODUCT_INFORMATION	PRODUCT_ID	Number	-	10	0	1	-	-	-
	PRODUCT_NAME	Varchar2	50	-	-	-	✓	-	-
	MODEL	Varchar2	50	-	-	-	✓	-	-
	PRICE	Number	-	10	0	-	✓	-	-
	CATEGORY	Varchar2	50	-	-	-	✓	-	-

The footer of the application shows 'Application Express 2.1.0.0.39' and 'Copyright © 1999, 2006, Oracle. All rights reserved.'

Create Table : Customer Information

Oracle Database Express Edition

User: DATABASE-FINAL

Home > SQL > SQL Commands

☒ Autocommit Display: 10000

Save **Run**

```
CREATE TABLE CUSTOMER_INFORMATION (
  CUSTOMER_ID NUMBER(10,0) PRIMARY KEY,
  CUSTOMER_NAME VARCHAR2(50),
  PHONE_NUMBER VARCHAR2(50),
  ORDER_ID NUMBER(10,0),
  EMAIL VARCHAR2(50)
);
DESC CUSTOMER_INFORMATION
ALTER TABLE CUSTOMER_INFORMATION ADD CONSTRAINT HASMANY_fk FOREIGN KEY (ORDER_ID) REFERENCES ORDER_DETAILS(ORDER_ID);
```

Results Explain Describe Saved SQL History

Object Type: **TABLE** Object: **CUSTOMER_INFORMATION**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER_INFORMATION	CUSTOMER_ID	Number	-	10	0	1	-	-	-
	CUSTOMER_NAME	Varchar2	50	-	-	-	✓	-	-
	PHONE_NUMBER	Varchar2	50	-	-	-	✓	-	-
	ORDER_ID	Number	-	10	0	-	✓	-	-
	EMAIL	Varchar2	50	-	-	-	✓	-	-
1 - 5									

Language: en-gb

Application Express 2.1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

Create Table: Order Details

Home > SQL > SQL Commands

☒ Autocommit Display: 10000

Save **Run**

```
CREATE TABLE ORDER_DETAILS (
  ORDER_ID NUMBER(10,0) PRIMARY KEY,
  ORDER_DATE DATE
);
DESC ORDER_DETAILS
```

Results Explain Describe Saved SQL History

Object Type: **TABLE** Object: **ORDER_DETAILS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ORDER_DETAILS	ORDER_ID	Number	-	10	0	1	-	-	-
	ORDER_DATE	Date	7	-	-	-	✓	-	-
1 - 2									

Language: en-gb

Application Express 2.1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

Create Table: Order Details 1

ORACLE Database Express Edition

User: DATABASE-FINAL

Home > SQL > SQL Commands

☒ Autocommit Display 10000

Save Run

```
CREATE TABLE ORDER_DETAILS_1 (  
  PRODUCT_ORDER_ID NUMBER(10,0) PRIMARY KEY,  
  PRODUCT_ID NUMBER(10,0),  
  ORDER_ID NUMBER(10,0)  
);  
DESC ORDER_DETAILS_1  
  
ALTER TABLE ORDER_DETAILS_1 ADD CONSTRAINT ORDER_fk FOREIGN KEY (PRODUCT_ID) REFERENCES PRODUCT_INFORMATION(PRODUCT_ID);  
ALTER TABLE ORDER_DETAILS_1 ADD CONSTRAINT ORDER1_fk FOREIGN KEY (ORDER_ID) REFERENCES ORDER_DETAILS(ORDER_ID);
```

Results Explain Describe Saved SQL History

Object Type TABLE Object ORDER_DETAILS_1

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ORDER_DETAILS_1	PRODUCT_ORDER_ID	Number	-	10	0	1	-	-	-
	PRODUCT_ID	Number	-	10	0	-	✓	-	-
	ORDER_ID	Number	-	10	0	-	✓	-	-
1 - 3									

Application Express 2 1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

Create Table : Product Details

Home > SQL > SQL Commands

☒ Autocommit Display 10000

Save Run

```
CREATE TABLE PRODUCT_DETAILS (  
  PRODUCT_ID NUMBER(10,0) PRIMARY KEY,  
  PRODUCT_NAME VARCHAR2(50) ,  
  MODEL VARCHAR2(50),  
  PRICE NUMBER(10,0),  
  CATEGORY VARCHAR2(50),  
  ORDER_DETAILS_ID NUMBER(10,0)  
);  
DESC PRODUCT_DETAILS  
ALTER TABLE PRODUCT_DETAILS ADD CONSTRAINT ORDER_fk FOREIGN KEY (ORDER_DETAILS_ID) REFERENCES PRODUCT_DETAILS_1(ORDER_DETAILS_ID);
```

Results Explain Describe Saved SQL History

Object Type TABLE Object PRODUCT_DETAILS

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PRODUCT_DETAILS	PRODUCT_ID	Number	-	10	0	1	-	-	-
	PRODUCT_NAME	Varchar2	50	-	-	-	✓	-	-
	MODEL	Varchar2	50	-	-	-	✓	-	-
	PRICE	Number	-	10	0	-	✓	-	-
	CATEGORY	Varchar2	50	-	-	-	✓	-	-
	ORDER_DETAILS_ID	Number	-	10	0	-	✓	-	-
1 - 6									

Application Express 2 1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

Create Table: Product Details 1

ORACLE Database Express Edition

User: DATABASE-FINAL

Home > SQL > SQL Commands

☒ Autocommit Display: 10000

Save Run

```
select * from PRODUCT_DETAILS_1

CREATE TABLE PRODUCT_DETAILS_1 (
  ORDER_DETAILS_ID NUMBER(10,0) PRIMARY KEY,
  QUANTITY NUMBER(10,0),
  PRODUCT VARCHAR2(50)
);
DESC PRODUCT_DETAILS_1
```

Results Explain Describe Saved SQL History

Object Type: TABLE Object: PRODUCT_DETAILS_1

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PRODUCT_DETAILS_1	ORDER_DETAILS_ID	Number	-	10	0	1	-	-	-
	QUANTITY	Number	-	10	0	-	✓	-	-
	PRODUCT	Varchar2	50	-	-	-	✓	-	-
1 - 3									

Language: en-gb

Application Express 2.1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

Create Table: Delivery

ORACLE Database Express Edition

User: DATABASE-FINAL

Home > SQL > SQL Commands

☒ Autocommit Display: 10000

Save Run

```
select * from DELIVERY

CREATE TABLE DELIVERY (
  ORDER_DETAILS_ORDER_ID NUMBER(10,0) PRIMARY KEY,
  ORDER_ID NUMBER(10,0),
  ORDER_DETAILS_ID NUMBER(10,0)
);
DESC DELIVERY
ALTER TABLE DELIVERY ADD CONSTRAINT ORDER25_fk FOREIGN KEY (ORDER_ID) REFERENCES ORDER_DETAILS(ORDER_ID);
ALTER TABLE DELIVERY ADD CONSTRAINT ORDER26_fk FOREIGN KEY (ORDER_DETAILS_ID) REFERENCES PRODUCT_DETAILS_1(ORDER_DETAILS_ID);
```

Results Explain Describe Saved SQL History

Object Type: TABLE Object: DELIVERY

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DELIVERY	ORDER_DETAILS_ORDER_ID	Number	-	10	0	1	-	-	-
	ORDER_ID	Number	-	10	0	-	✓	-	-
	ORDER_DETAILS_ID	Number	-	10	0	-	✓	-	-
1 - 3									

Language: en-gb

Application Express 2.1.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

Inserted Values in the tables

Data insertion in Customer Details table:

```
CREATE SEQUENCE ZX
START WITH 101
INCREMENT BY 1
MAXVALUE 110
NOCYCLE;

INSERT INTO CUSTOMER_DETAILS (CUSTOMER_ID, CUSTOMER_NAME, PHONE_NUMBER, PRODUCT_ID, EMAIL)
VALUES (ZX.NEXTVAL, 'Zayed Khan', '01345476543', 201, 'zayed@gmail.com');
SELECT * FROM CUSTOMER_DETAILS
```

Results Explain Describe Saved SQL History

CUSTOMER_ID	CUSTOMER_NAME	PHONE_NUMBER	PRODUCT_ID	EMAIL
101	Afnan Nahin	01712876543	111	afnan@gmail.com
102	Aman Ullah	01612876543	122	aman@gmail.com
103	Hero Alom	01812876543	111	hero@gmail.com
104	Sharuk khan	01912876543	155	sri@gmail.com
105	Salman Khan	01212876543	155	bhajan@gmail.com
106	Tom Cruise	01345876543	166	tom@gmail.com
107	Anade Armas	01894876543	177	anade@gmail.com
108	Leo Messi	01973176543	188	messi@gmail.com
109	Katherine Langford	01542876543	201	Katherine@gmail.com
110	Zayed Khan	01345476543	201	zayed@gmail.com

10 rows returned in 0.00 seconds CSV Export

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

Data insertion in Product Information table:

☒ Autocommit Display 10000 Save Run

```
INSERT INTO PRODUCT_INFORMATION (PRODUCT_ID, PRODUCT_NAME, MODEL, PRICE, CATEGORY)
VALUES (177, 'Camara', 'GoPro', 50000, 'GoPro Hero10');

INSERT INTO PRODUCT_INFORMATION (PRODUCT_ID, PRODUCT_NAME, MODEL, PRICE, CATEGORY)
VALUES (188, 'VideoGames', 'Sony PS5', 65000, 'PS5');

INSERT INTO PRODUCT_INFORMATION (PRODUCT_ID, PRODUCT_NAME, MODEL, PRICE, CATEGORY)
VALUES (199, 'Power Bank', 'MagSafe Battery', 12000, 'Battery Pack');

INSERT INTO PRODUCT_INFORMATION (PRODUCT_ID, PRODUCT_NAME, MODEL, PRICE, CATEGORY)
VALUES (201, 'Drone', 'DJI AVATA 2', 132000, 'VIDEO SHOOTING');

select * from PRODUCT_INFORMATION
```

Results Explain Describe Saved SQL History

PRODUCT_ID	PRODUCT_NAME	MODEL	PRICE	CATEGORY
111	Lap Top	MacBook	120000	Macbookair M1
122	Phone	IPhone	150000	15 Pro Max
133	Watch	Galaxy Watch6	50000	SmartWatch
144	Speaker	JBL	37000	BoobBox2
155	TV	Sony Bravia	125000	55-inch LED HDR 4K Ultra
166	AirPods	Apple	68000	AirPodsMax
177	Camara	GoPro	50000	GoPro Hero10
188	VideoGames	Sony PS5	65000	PS5
199	Power Bank	MagSafe Battery	12000	Battery Pack
201	Drone	DJI AVATA 2	132000	VIDEO SHOOTING

10 rows returned in 0.00 seconds CSV Export

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

Data insertion in Customer Information table:

☒ Autocommit Display 10000 Save Run

```
CREATE SEQUENCE CV
START WITH 101
INCREMENT BY 1
MAXVALUE 110
NOCYCLE;
INSERT INTO CUSTOMER_INFORMATION (CUSTOMER_ID, CUSTOMER_NAME, PHONE_NUMBER, ORDER_ID, EMAIL)
VALUES (CV.NEXTVAL, 'Zayed Khan', '01345476543', 232, 'zayed@gmail.com');
SELECT * FROM CUSTOMER_INFORMATION
```

Results Explain Describe Saved SQL History

CUSTOMER_ID	CUSTOMER_NAME	PHONE_NUMBER	ORDER_ID	EMAIL
101	Afnan Nahin	01712876543	223	afnan@gmail.com
102	Aman Ullah	01612876543	224	aman@gmail.com
103	Hero Alom	01812876543	225	hero@gmail.com
104	Shrukh Khan	01912876543	226	srk@gmail.com
105	Salman Khan	01212876543	227	bhaihan@gmail.com
106	Tom Cruise	01345876543	228	tom@gmail.com
107	Anade Armas	01894876543	229	anade@gmail.com
108	Leo Messi	01973176543	230	messi@gmail.com
109	Katherine Langford	01542876543	231	Katherine@gmail.com
110	Zayed Khan	01345476543	232	zayed@gmail.com

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

Language: en-gb Application Express 2.1.0.00.39
Copyright © 1999, 2005, Oracle. All rights reserved.

Data insertion in Order details table:

Home > SQL > **SQL Commands**

☒ Autocommit Display 10000 Save Run

```
select * from ORDER_DETAILS
CREATE SEQUENCE BN
START WITH 223
INCREMENT BY 1
MAXVALUE 232
NOCYCLE;
INSERT INTO ORDER_DETAILS (ORDER_ID, ORDER_DATE)
VALUES (BN.NEXTVAL, TO_DATE('9-JAN-24', 'DD-MON-YY'));
```

Results Explain Describe Saved SQL History

ORDER_ID	ORDER_DATE
223	23-JAN-24
224	13-MAY-24
225	22-FEB-24
226	28-MAR-24
227	30-JAN-24
228	23-APR-24
229	14-MAY-24
230	02-JAN-24
231	03-JAN-24
232	09-JAN-24

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

Data insertion in Order details 1 table:

Home > SQL > SQL Commands

☒ Autocommit Display 10000 Save Run

```
CREATE SEQUENCE LK
START WITH 1
INCREMENT BY 1
MAXVALUE 10
NOCYCLE;
SELECT * FROM ORDER_DETAILS_1;
INSERT INTO ORDER_DETAILS_1 (PRODUCT_ORDER_ID,PRODUCT_ID,ORDER_ID)
VALUES (LK.NEXTVAL,201,232);
```

Results Explain Describe Saved SQL History

PRODUCT_ORDER_ID	PRODUCT_ID	ORDER_ID
1	111	223
2	122	224
3	111	225
4	155	226
5	155	227
6	166	228
7	177	229
8	188	230
9	201	231
10	201	232

10 rows returned in 0.01 seconds [CSV Export](#)

Data insertion in product details table:

Home > SQL > SQL Commands

☒ Autocommit Display 10000 Save Run

```
VALUES (199,'Power Bank','MagSafe Battery',12000,'Battery Pack',52);
INSERT INTO PRODUCT_DETAILS (PRODUCT_ID, PRODUCT_NAME,MODEL,PRICE,CATEGORY,ORDER_DETAILS_ID)
VALUES (201,'Drone','DJI AVATA 2',132000,'VIDEO SHOOTING',53);
select * from PRODUCT_DETAILS
```

Results Explain Describe Saved SQL History

PRODUCT_ID	PRODUCT_NAME	MODEL	PRICE	CATEGORY	ORDER_DETAILS_ID
111	LapTop	MacBook	120000	Macbookair M1	44
122	Phone	IPhone	150000	15 Pro Max	45
133	Watch	Galaxy Watch6	50000	SmartWatch	46
144	Speaker	JBL	37000	BoobBox2	47
155	TV	Sony Bravia	125000	55-inch LED HDR 4K Ultra	48
166	AirPods	Apple	68000	AirPodsMax	49
177	Camera	GoPro	50000	GoPro Hero10	50
188	VideoGames	Sony PS5	65000	PSS	51
199	Power Bank	MagSafe Battery	12000	Battery Pack	52
201	Drone	DJI AVATA 2	132000	VIDEO SHOOTING	53

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

Data insertion in Product details 1 table:

Home > SQL > SQL Commands

☒ Autocommit Display 10000 Save Run

```
select * from PRODUCT_DETAILS_1
INSERT INTO PRODUCT_DETAILS_1 (ORDER_DETAILS_ID,QUANTITY,PRODUCT)
VALUES (44,3,'lapTop');
INSERT INTO PRODUCT_DETAILS_1 (ORDER_DETAILS_ID,QUANTITY,PRODUCT )
VALUES (45,2,'Phone');
INSERT INTO PRODUCT_DETAILS_1 (ORDER_DETAILS_ID,QUANTITY,PRODUCT )
VALUES (46,0,'Watch');
INSERT INTO PRODUCT_DETAILS_1 (ORDER_DETAILS_ID,QUANTITY,PRODUCT )
VALUES (47,0,'Speaker');
INSERT INTO PRODUCT_DETAILS_1 (ORDER_DETAILS_ID,QUANTITY,PRODUCT )
VALUES (48,4,'TV');
INSERT INTO PRODUCT_DETAILS_1 (ORDER_DETAILS_ID,QUANTITY,PRODUCT )
```

Results Explain Describe Saved SQL History

ORDER_DETAILS_ID	QUANTITY	PRODUCT
44	3	lapTop
45	2	Phone
46	0	Watch
47	0	Speaker
48	4	TV
49	3	AirPods
50	2	Camara
51	4	VideoGames
52	0	Power Bank
53	6	Drone

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

Data insertion in Delivery table:

Home > SQL > SQL Commands

☒ Autocommit Display 10000 Save Run

```
VALUES (85,227,48);
INSERT INTO DELIVERY ( ORDER_DETAILS_ORDER_ID, ORDER_ID,ORDER_DETAILS_ID)
VALUES (86,228,49);
INSERT INTO DELIVERY ( ORDER_DETAILS_ORDER_ID, ORDER_ID,ORDER_DETAILS_ID)
VALUES (87,229,50);
INSERT INTO DELIVERY ( ORDER_DETAILS_ORDER_ID, ORDER_ID,ORDER_DETAILS_ID)
VALUES (88,230,51);
INSERT INTO DELIVERY ( ORDER_DETAILS_ORDER_ID, ORDER_ID,ORDER_DETAILS_ID)
VALUES (89,231,53);
INSERT INTO DELIVERY ( ORDER_DETAILS_ORDER_ID, ORDER_ID,ORDER_DETAILS_ID)
VALUES (90,232,53);
select * from DELIVERY
```

Results Explain Describe Saved SQL History

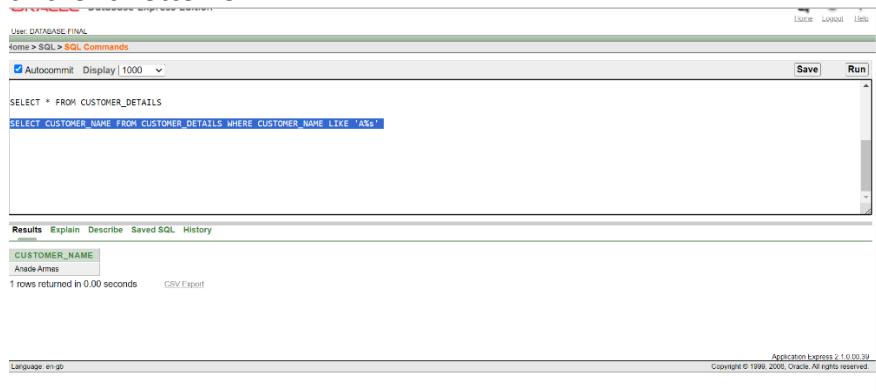
ORDER_DETAILS_ORDER_ID	ORDER_ID	ORDER_DETAILS_ID
81	223	44
82	224	45
83	225	44
84	226	48
85	227	48
86	228	49
87	229	50
88	230	51
89	231	53
90	232	53

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

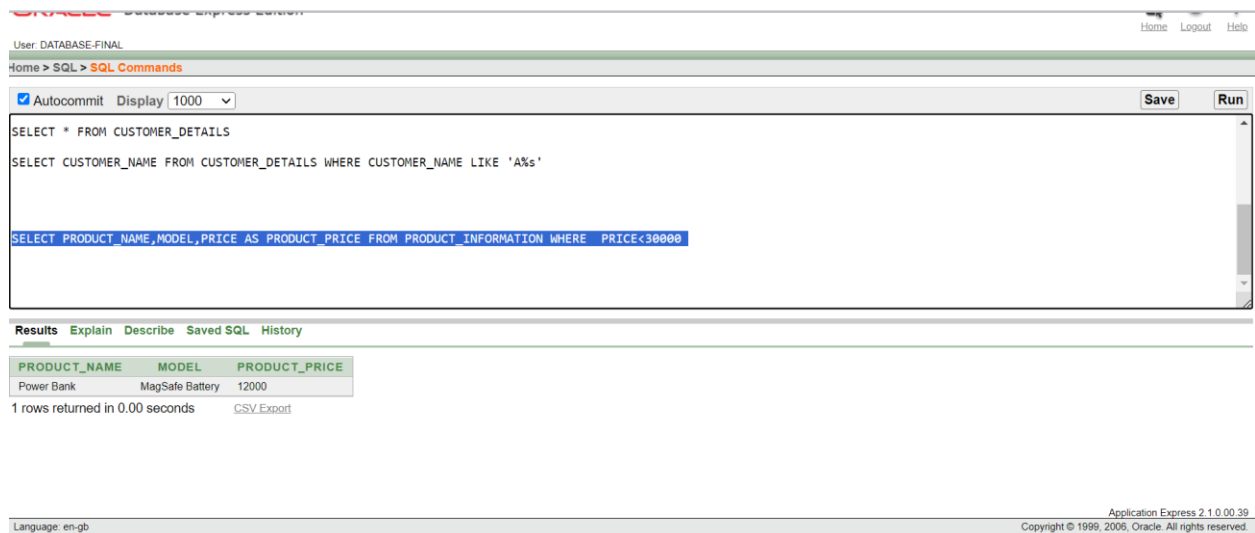
Query Test in DB

CONDITIONAL STATEMENT :

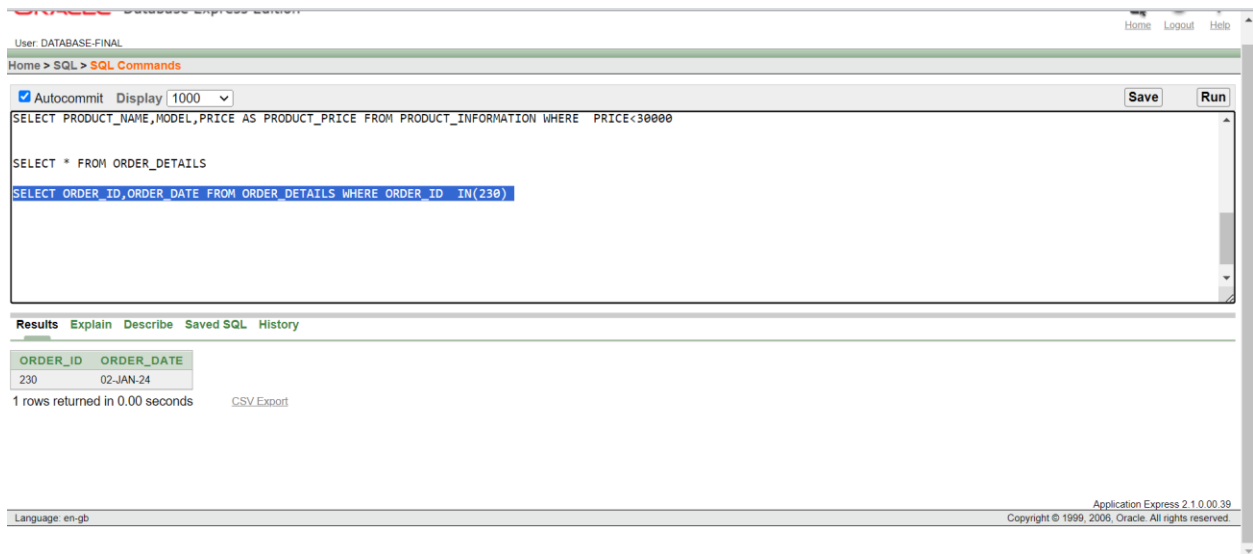
*Display the customer name from Customer Details where the name starting letter 'A' and end letter 's'



*Display customer name,model and price from product information where price is less than 30000



Display Order id and Order date from order details where order id is 230



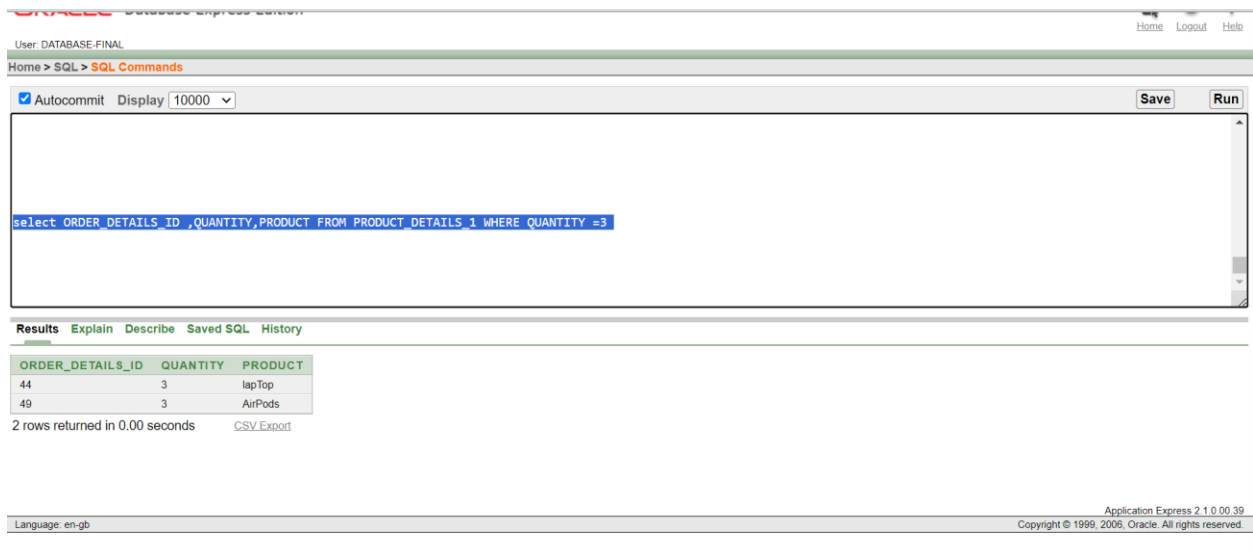
The screenshot shows the Oracle SQL Developer interface. The top bar indicates the user is 'DATABASE-FINAL'. The main window is titled 'SQL Commands' and contains the following SQL query:

```
SELECT * FROM ORDER_DETAILS  
SELECT ORDER_ID, ORDER_DATE FROM ORDER_DETAILS WHERE ORDER_ID IN(230)
```

The query has been executed, and the results are displayed in a table with two columns: 'ORDER_ID' and 'ORDER_DATE'. The table contains one row with the values '230' and '02-JAN-24'. Below the table, it states '1 rows returned in 0.00 seconds'. The interface also includes a 'Results' tab, a 'CSV Export' button, and a footer with 'Language: en-gb' and 'Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.'

ORDER_ID	ORDER_DATE
230	02-JAN-24

Display Order_details_id ,quantity, product from product_details_1 where quantity is equal to 3



The screenshot shows the Oracle SQL Developer interface. The top bar indicates the user is 'DATABASE-FINAL'. The main window is titled 'SQL Commands' and contains the following SQL query:

```
select ORDER_DETAILS_ID ,QUANTITY,PRODUCT FROM PRODUCT_DETAILS_1 WHERE QUANTITY =3
```

The query has been executed, and the results are displayed in a table with three columns: 'ORDER_DETAILS_ID', 'QUANTITY', and 'PRODUCT'. The table contains two rows with the values '44', '3', 'lapTop' and '49', '3', 'AirPods'. Below the table, it states '2 rows returned in 0.00 seconds'. The interface also includes a 'Results' tab, a 'CSV Export' button, and a footer with 'Language: en-gb' and 'Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.'

ORDER_DETAILS_ID	QUANTITY	PRODUCT
44	3	lapTop
49	3	AirPods

Display product id,product name,price ,category and model from product details where order_details_id is between 43 to 47

The screenshot shows the Oracle SQL Developer interface. At the top, the user is 'DATABASE-FINAL'. The main window displays a SQL query: `SELECT ORDER_DETAILS_ID,QUANTITY,PRODUCT FROM PRODUCT_DETAILS_1 WHERE QUANTITY=3`. Below the query, there is a table with 4 rows of results. The table has columns: PRODUCT_ID, PRODUCT_NAME, PRICE, CATEGORY, and MODEL. The data rows are: (111, LapTop, 120000, Macbookair M1, MacBook), (122, Phone, 150000, 15 Pro Max, iPhone), (133, Watch, 50000, SmartWatch, Galaxy Watch6), and (144, Speaker, 37000, BoobBox2, JBL). The status bar at the bottom indicates '4 rows returned in 0.00 seconds' and 'CSV Export'.

PRODUCT_ID	PRODUCT_NAME	PRICE	CATEGORY	MODEL
111	LapTop	120000	Macbookair M1	MacBook
122	Phone	150000	15 Pro Max	iPhone
133	Watch	50000	SmartWatch	Galaxy Watch6
144	Speaker	37000	BoobBox2	JBL

SINGLE ROW FUNCTION:

*Change the product name from product information in capital letter

The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL query: `SELECT PRODUCT_ID,PRODUCT_NAME,PRICE,CATEGORY,MODEL FROM PRODUCT_DETAILS WHERE ORDER_DETAILS_ID BETWEEN 43 AND 47`. Below the query, there is a table with 10 rows of results. The table has columns: PRODUCT_NAME and UPPER(PRODUCT_NAME). The data rows are: (LapTop, LAPTOP), (Phone, PHONE), (Watch, WATCH), (Speaker, SPEAKER), (TV, TV), (AirPods, AIRPODS), (Camara, CAMARA), (VideoGames, VIDEOGAMES), (Power Bank, POWER BANK), and (Drone, DRONE). The status bar at the bottom indicates '10 rows returned in 0.01 seconds' and 'CSV Export'.

PRODUCT_NAME	UPPER(PRODUCT_NAME)
LapTop	LAPTOP
Phone	PHONE
Watch	WATCH
Speaker	SPEAKER
TV	TV
AirPods	AIRPODS
Camara	CAMARA
VideoGames	VIDEOGAMES
Power Bank	POWER BANK
Drone	DRONE

*Find the length of customer name from customer details

```
SELECT PRODUCT_ID,PRODUCT_NAME,PRICE,CATEGORY,MODEL FROM PRODUCT_DETAILS WHERE ORDER_DETAILS_ID BETWEEN 43 AND 47
```

```
SELECT CUSTOMER_NAME,LENGTH(CUSTOMER_NAME) FROM CUSTOMER_DETAILS
```

Results Explain Describe Saved SQL History

CUSTOMER_NAME	LENGTH(CUSTOMER_NAME)
Afnan Nahin	11
Aman Ullah	10
Hero Alom	9
Sharuk khan	11
Salman Khan	11
Tom Cruise	10
Anade Armas	11
Leo Messi	9
Katherine Langford	18
Zayed Khan	10

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

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

*Show the concat between customer name and product id from customer details

```
SELECT PRODUCT_ID,PRODUCT_NAME,PRICE,CATEGORY,MODEL FROM PRODUCT_DETAILS WHERE ORDER_DETAILS_ID BETWEEN 43 AND 47
```

```
SELECT CUSTOMER_NAME,PRODUCT_ID,CONCAT(CUSTOMER_NAME , PRODUCT_ID) AS JOIN FROM CUSTOMER_DETAILS
```

Results Explain Describe Saved SQL History

CUSTOMER_NAME	PRODUCT_ID	JOIN
Afnan Nahin	111	Afnan Nahin111
Aman Ullah	122	Aman Ullah122
Hero Alom	133	Hero Alom133
Sharuk khan	144	Sharuk khan144
Salman Khan	155	Salman Khan155
Tom Cruise	166	Tom Cruise166
Anade Armas	177	Anade Armas177
Leo Messi	188	Leo Messi188
Katherine Langford	199	Katherine Langford199
Zayed Khan	201	Zayed Khan201

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

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

*Print the first 4 letter of product name from product details

The screenshot shows the SQL Developer interface. The query editor contains the following SQL statement:

```
SELECT PRODUCT_NAME, SUBSTR(PRODUCT_NAME, 1, 4) AS SUBSTR FROM PRODUCT_DETAILS
```

The results pane displays a table with two columns: **PRODUCT_NAME** and **SUBSTR**. The data is as follows:

PRODUCT_NAME	SUBSTR
LapTop	LapT
Phone	Phon
Watch	Watc
Speaker	Spea
TV	TV
AirPods	AirP
Camera	Cama
VideoGames	Vide
Power Bank	Powe
Drone	Dron

10 rows returned in 0.00 seconds

*Search the category from product information where product name is TV and find the location of 'H'

The screenshot shows the SQL Developer interface. The query editor contains the following SQL statement:

```
SELECT CATEGORY, INSTR(CATEGORY, 'H') AS INSTR FROM PRODUCT_INFORMATION WHERE PRODUCT_NAME = 'TV'
```

The results pane displays a table with two columns: **CATEGORY** and **INSTR**. The data is as follows:

CATEGORY	INSTR
55-inch LED HDR 4K Ultra	13

1 rows returned in 0.00 seconds

MULTIPLE ROW FUNCTION :

*Display the max and min price from product information

The screenshot shows the Oracle SQL Developer interface. At the top, the user is 'DATABASE-FINAL'. The breadcrumb navigation is 'Home > SQL > SQL Commands'. The 'Autocommit' checkbox is checked, and the 'Display' size is set to 1000. The SQL command entered is: `SELECT CATEGORY, INSTR(CATEGORY, 'H') AS INSTR FROM PRODUCT_INFORMATION WHERE PRODUCT_NAME = 'TV'`. Below this, the query is executed, and the results are displayed in a table with two columns: 'MAX(PRICE)' and 'MIN(PRICE)'. The values are 150000 and 12000 respectively. The status bar indicates '1 rows returned in 0.00 seconds' and provides a 'CSV Export' link. The footer shows 'Language: en-gb' and 'Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved.'

MAX(PRICE)	MIN(PRICE)
150000	12000

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

*Display the sum price from product information

The screenshot shows the Oracle SQL Developer interface. At the top, the user is 'DATABASE-FINAL'. The breadcrumb navigation is 'Home > SQL > SQL Commands'. The 'Autocommit' checkbox is checked, and the 'Display' size is set to 1000. The SQL command entered is: `SELECT CATEGORY, INSTR(CATEGORY, 'H') AS INSTR FROM PRODUCT_INFORMATION WHERE PRODUCT_NAME = 'TV'`. Below this, the query is executed, and the results are displayed in a table with one column: 'SUM(PRICE)'. The value is 809000. The status bar indicates '1 rows returned in 0.00 seconds' and provides a 'CSV Export' link. The footer shows 'Language: en-gb' and 'Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved.'

SUM(PRICE)
809000

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

*Display the avg price from product details

ORACLE Database Express Edition

User: DATABASE-FINAL

Home > SQL > SQL Commands

☒ Autocommit Display 1000 Save Run

```
SELECT AVG(PRICE) FROM PRODUCT_DETAILS
```

Results Explain Describe Saved SQL History

AVG(PRICE)
80900

1 rows returned in 0.00 seconds CSV Export

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

SUBQUERY:

*Display the customer name,phone number from customer details where product id is greater than Leo Messi product id

ORACLE Database Express Edition

User: DATABASE-FINAL

Home > SQL > SQL Commands

☒ Autocommit Display 1000 Save Run

```
SELECT CUSTOMER_NAME,PHONE_NUMBER FROM CUSTOMER_DETAILS WHERE PRODUCT_ID > (SELECT PRODUCT_ID FROM CUSTOMER_DETAILS WHERE CUSTOMER_NAME = 'Leo Messi')  
SELECT * FROM CUSTOMER_DETAILS
```

Results Explain Describe Saved SQL History

CUSTOMER_NAME	PHONE_NUMBER
Katherine Langford	01542876543
Zayed Khan	01345476543

2 rows returned in 0.00 seconds CSV Export

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

*Display the product name,model from product details where price is greater than AirPodsMax price

User: DATABASE-FINAL

Home > SQL > SQL Commands

☒ Autocommit Display 1000 Save Run

```
SELECT CUSTOMER_NAME,PHONE_NUMBER FROM CUSTOMER_DETAILS WHERE PRODUCT_ID > (SELECT PRODUCT_ID FROM CUSTOMER_DETAILS WHERE CUSTOMER_NAME = 'Leo Messi')
SELECT * FROM CUSTOMER_DETAILS
SELECT PRODUCT_NAME,MODEL FROM PRODUCT_DETAILS WHERE PRICE > (SELECT PRICE FROM PRODUCT_DETAILS WHERE CATEGORY = 'AirPodsMax')
SELECT * FROM PRODUCT_DETAILS
```

Results Explain Describe Saved SQL History

PRODUCT_NAME	MODEL
LapTop	MacBook
Phone	iPhone
TV	Sony Bravia
Drone	DJI AVATA 2

4 rows returned in 0.01 seconds [CSV Export](#)

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

JOINING:

*Show an equijoin from two table where it's possible

```
SELECT CUSTOMER_NAME FROM CUSTOMER_DETAILS WHERE PRODUCT_ID>(SELECT PRODUCT_ID FROM CUSTOMER_DETAILS WHERE CUSTOMER_NAME ='Leo Messi')
SELECT P.PRODUCT_ID,P.PRODUCT_NAME,P.PRICE,P.CATEGORY,P.MODEL,Q.QUANTITY FROM PRODUCT_DETAILS P,PRODUCT_DETAILS_1 Q WHERE
P.ORDER_DETAILS_ID=Q.ORDER_DETAILS_ID
```

Results Explain Describe Saved SQL History

PRODUCT_ID	PRODUCT_NAME	PRICE	CATEGORY	MODEL	QUANTITY
111	LapTop	120000	Macbookair M1	MacBook	3
122	Phone	150000	15 Pro Max	iPhone	2
133	Watch	50000	SmartWatch	Galaxy Watch6	0
144	Speaker	37000	BoobBox2	JBL	0
155	TV	125000	55-inch LED HDR 4K Ultra	Sony Bravia	4
166	AirPods	68000	AirPodsMax	Apple	3
177	Camara	50000	GoPro Hero10	GoPro	2
188	VideoGames	65000	PS5	Sony PS5	4
199	Power Bank	12000	Battery Pack	MagSafe Battery	0
201	Drone	132000	VIDEO SHOOTING	DJI AVATA 2	6

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

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

*Show an non-equijoin from two table where it's possible

Autocommit Display 1000 Save Run

```

SELECT C.CUSTOMER_NAME,C.EMAIL,A.QUANTITY FROM CUSTOMER_DETAILS C,PRODUCT_DETAILS_1 A WHERE A.ORDER_DETAILS_ID BETWEEN C.CUSTOMER_ID AND C.PRODUCT_ID
SELECT * FROM PRODUCT_INFORMATION
SELECT * FROM DELIVERY
SELECT C.PRODUCT_NAME,C.MODEL,A.ORDER_DETAILS_ORDER_ID FROM PRODUCT_INFORMATION C,DELIVERY A WHERE C.PRODUCT_ID BETWEEN A.ORDER_DETAILS_ID AND A.ORDER_ID

```

Results Explain Describe Saved SQL History

PRODUCT_NAME	MODEL	ORDER_DETAILS_ORDER_ID
LapTop	MacBook	81
Phone	iPhone	81
Watch	Galaxy Watch6	81
Speaker	JBL	81
TV	Sony Bravia	81
AirPods	Apple	81
Camara	GoPro	81
VideoGames	Sony PS5	81
Power Bank	MagSafe Battery	81
Drone	DJI AVATA 2	81
LapTop	MacBook	82
Phone	iPhone	82

*Show an outer join from two table where it's possible

Home > SQL > SQL Commands Autocommit Display 1000 Save Run

```

SELECT C.CUSTOMER_NAME,c.EMAIL,P.PRODUCT_ID FROM CUSTOMER_DETAILS C,PRODUCT_INFORMATION P WHERE C.PRODUCT_ID(+) =P.PRODUCT_ID

select* from CUSTOMER_DETAILS
select* from PRODUCT_INFORMATION

```

Results Explain Describe Saved SQL History

CUSTOMER_NAME	EMAIL	PRODUCT_ID
Afhan Nahn	afhan@gmail.com	111
Aman Ullah	aman@gmail.com	122
Hero Alam	hero@gmail.com	111
Sharuk Khan	srk@gmail.com	155
Salman Khan	bhagyan@gmail.com	155
Tom Cruise	tom@gmail.com	166
Anade Armas	anade@gmail.com	177
Leo Messi	messi@gmail.com	188
Katherine Langford	Katherine@gmail.com	201
Zayed Khan	zayed@gmail.com	201
-	-	144
-	-	133
-	-	199

13 rows returned in 0.00 seconds CSV Export

VIEW:

*Create a view where name is AB

The screenshot shows the SQL Developer interface with the 'SQL Commands' window. The 'Autocommit' checkbox is checked, and the 'Display' size is set to 1000. The SQL script in the editor is as follows:

```
DESC CUSTOMER_DETAILS
DROP TABLE CUSTOMER_DETAILS
select * from CUSTOMER_DETAILS

CREATE VIEW AB
AS SELECT CUSTOMER_ID,PHONE_NUMBER FROM CUSTOMER_DETAILS
SELECT * FROM AB
```

Below the editor, the 'Results' tab is selected, showing the message 'View created.' and a duration of '0.00 seconds'. The status bar at the bottom indicates 'Language: en-gb' and 'Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved.'

*Create a view where name is AB

The screenshot shows the SQL Developer interface with the 'SQL Commands' window. The 'Autocommit' checkbox is checked, and the 'Display' size is set to 1000. The SQL script in the editor is as follows:

```
DESC CUSTOMER_DETAILS
DROP TABLE CUSTOMER_DETAILS
select * from CUSTOMER_DETAILS

CREATE VIEW AB
AS SELECT CUSTOMER_ID,PHONE_NUMBER FROM CUSTOMER_DETAILS
SELECT * FROM AB

DESC AB
```

Below the editor, the 'Describe' tab is selected, showing the object type 'VIEW Object AB'. The table structure is as follows:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
AB	CUSTOMER_ID	Number	-	10	0	-	-	-	-
	PHONE_NUMBER	Varchar2	50	-	-	-	✓	-	-
									1 - 2

The status bar at the bottom indicates 'Language: en-gb' and 'Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved.'

*The data of customer details after creating view

The screenshot shows the SQL Developer interface with the 'SQL Commands' window. The 'Autocommit' checkbox is checked, and the 'Display' size is set to 1000. The SQL script in the editor is as follows:

```
DROP TABLE CUSTOMER_DETAILS
select * from CUSTOMER_DETAILS

CREATE VIEW AB
AS SELECT CUSTOMER_ID,PHONE_NUMBER FROM CUSTOMER_DETAILS
SELECT * FROM AB
```

Below the editor, the 'Results' tab is selected, showing the data of the view 'AB'. The data is as follows:

CUSTOMER_ID	PHONE_NUMBER
101	01712876543
102	01612876543
103	01812876543
104	01912876543
105	01212876543
106	01345676543
107	01894876543
108	01973176543
109	01542876543
110	01345476543

Below the table, it says '10 rows returned in 0.00 seconds' and 'CSV Export'. The status bar at the bottom indicates 'Language: en-gb' and 'Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved.'

RELATIONAL ALGEBRA :

- **Display the customer name from Customer Details where the name starting letter 'A' and end letter 's'**
----- Π customer_name $[\sigma \text{ c_name Like 'A\%' (Customer Details)}]$
- **Display customer name,model and price from product information where price is less than 3000**
----- Π customer_name,model,price $[\sigma \text{ price } < 3000 \text{ (Product Information)}]$
- **Display Order id and Order date from order details where order id is 230**
----- Π order_id,order_date $[\sigma \text{ order_id} = 230 \text{ (Order Details)}]$
- **Display Order_details_id ,quantity, product from product_details_1 where quantity is equal to 3**
----- Π order_details_id,quantity,product $[\sigma \text{ quantity} = 3 \text{ (Product Details)}]$
- **Display product id,product name,price ,category and model from product details where order_details_id is between 43 to 47**
----- Π product_id,product_name,model,price,category $[\sigma \text{ order_details_id between 43 and 47 (Product Details)}]$
- **Display the product name,model from product details where price is greater than AirPodsMax price**
----- Π product_name,model $[\sigma \text{ price} > \text{AirpodsMax price (Product Details)}]$
- **Display the customer name,phone number from customer details where product id is greater than Leo Messi product id**
----- Π customer_name,phone_number $[\sigma \text{ product_id} > \text{LeoMessi product_id (Customer Details)}]$
- **Display customer name,email,phone number from customer details**
----- Π customer_name,email,phone_number (Customer Details)
- **Display customer name,model and price from product information where price is GREATERthan 50000**
----- Π customer_name,model,price $[\sigma \text{ price } > 50000 \text{ (Product Information)}]$
- **Display the customer name,phone number from customer details where product id is greater than Anade Armas product id**
----- Π customer_name,phone_number $[\sigma \text{ product_id} > \text{AnadeArmas product_id (Customer Details)}]$

Conclusion:

In summary, the Gadget and Gear Management System (GGMS) project has successfully streamlined the purchasing process for electronic devices. By offering a diverse range of products and implementing centralized order management, GGMS has significantly enhanced operational efficiency and customer satisfaction. Moving forward, GGMS remains dedicated to innovation and efficiency in gadget and gear management.