



AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

FACULTY OF SCIENCE & TECHNOLOGY

INTRODUCTION TO DATABASE

Fall 2023-24

Project Name: Preloved Products Management System.

Section: C

Supervised By

JUENA AHMED NOSHIN

Assistant Professor

Submitted By

Name	ID	Contribution
1) S. S. Zobaer Ahmed	22-49415-3	20%
2) Efaz Mashrur Amin	22-49393-3	20%
3) S.M.Abid Hossain Mahid	22-49419-3	20%
4) Nazmul Hassan Nayeem	22-49385-3	20%
5) MD.Thiyab Rahman	23-50838-1	20%

Date of Submission: 2nd January 2024.

CONTENT PAGE

No	Topic	Page
01	Introduction	03
02	Scenario Description	03
03	ER Diagram	04
04	Normalization	04-06
05	Schema Diagram	07
06	Table Creation	08-12
07	Data Insertion	13-17
08	Query Writing	18-24
09	Relational Algebra	25
10	DB Connection	26
11	Conclusion	27

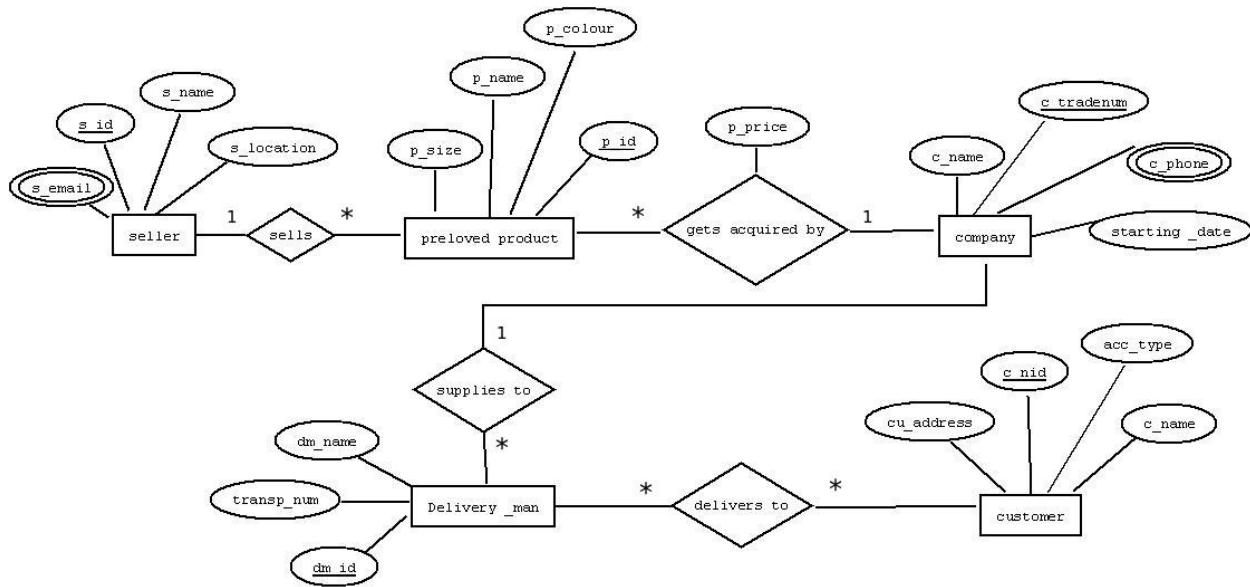
INTRODUCTION:

In a world driven by consumerism and sustainability, the management of preloved products has emerged as a pivotal challenge and opportunity. The Preloved Products Management System stands at the intersection of technological innovation and conscientious consumption, offering a sophisticated solution to streamline the lifecycle of second-hand goods. As we navigate the dynamic landscape of e-commerce and environmental consciousness, an efficient and robust database management system becomes the cornerstone for orchestrating the seamless flow of information, ensuring the optimal functionality of a Preloved Products Management System. This journey delves into the realms of database architecture, data integrity, and user-centric design, converging to empower businesses and individuals alike in their quest for sustainable commerce.

Scenario Description:

In a Preloved Products Management System, a seller can sell some preloved products. A seller can sell many products. A seller has name, id, email address, location. The products have size, color, id, name. Many products get acquired by only one company. A company has a trade number, phone number, starting date. A company hire many deliveries man. The delivery man has also name, id, and transport number. The delivery man's delivers the products to many customers. The delivery man identifies to their customers by name and NID. The delivery man has the access of the customer's address and account type.

ER DIAGRAM:



Normalization:

Sells (**s_id**, **s_location**, **s_name**, **s_email**, **p_id**, **p_name**, **p_price**, **p_size**, **p_colour**)

1NF: **s_email** is a multivalued attribute.

2NF: **s_id**, **s_name**, **s_location**, **s_email**
p_id, **p_name**, **p_color**, **p_size**, **p_price**

3NF: **s_id**, **s_name**, **s_location**, **s_email**
p_sizeid, **p_size**, **p_price**, **p_color**
p_id, **p_name**

Table for Sells:

1. **s_id**, **s_name**, **s_location**
2. **p_sizeid**, **p_size**, **p_price**, **p_color**
3. **p_id**, **p_name**, **p_sizeid**, **s_id**
4. **s_id**, **s_email** – composite primary key

gets acquired by (p_id, p_size, p_name, p_colour, p_price, c_name, c_tradeNum, c_phone, starting_date)

1NF: c_phone is a multivalued attribute

2NF: p_id, p_name, p_size, p_color, p_price
c_tradeNum, c_name, c_phone, starting_date

3NF: p_id, p_name
p_sizeid, p_size, p_price, p_color
c_tradeNum, c_name, c_phone, starting_date

Table for gets acquired by:

1. p_id, p_name, p_price, c_tradeNum, p_sizeid
2. p_sizeid, p_price, p_size, p_color
3. c_tradeNum, c_name, starting_date
4. c_tradeNum, c_phone-composite primary key

Supplies to (s_tradeNum, s_name, s_phone, starting_date, p_price, dm_id, transp_num, dm_name)

1NF: s_phone is a multivalued attribute.

2NF: s_tradeNum, s_name, s_phone, p_price, starting_date
dm_id, dm_name, transp_num

3NF: No transitive dependency

- s_tradeNum, s_name, s_phone, p_price, starting_date
dm_id, dm_name, transp_num

Table for Supplies to:

1. s_tradeNum, s_name, p_price, starting_date
2. dm_id, dm_name, transp_num, s_tradeNum
3. s_tradeNum, s_phone- composite primary key

Delivers to (dm_id, transp_num, dm_name, c_nid, cu_address, acc_type, c_name)

1NF: No multivalued Attribute

2NF: **dm_id**, dm_name, transp_num
c_nid, cu_address, acc_type, c_name

3NF: No transitive dependency

dm_id, dm_name, transp_num
c_nid, cu_address, acc_type, c_name

Table for delivers to:

1. **dm_id**, dm_name, transp_num
2. **c_nid**, cu_address, acc_type, c_name
3. **t_id**, **dm_id**, **c_nid**

Final Table List

1. **s_id**, d_name, d_location – **Seller**
2. **p_id**, p_name, **p_sizeid**, **s_id** – **Preloved Product**
3. **p_sizeid**, p_size, p_price, p_color – **Product Information**
4. **s_id**, s_email – **Seller1**
5. **p_id**, p_name, p_price, **c_tradeNum**, **p_sizeid** – **Product1**
6. **c_tradeNum**, s_name, p_price, starting_date - **Company**
7. **c_tradeNum**, s_phone – **Company1**
8. **dm_id**, dm_name, transp_num, **c_tradeNum** – **DeliveryMan**
9. **c_nid**, cu_address, acc_type, c_name - **Customer**

10. **t_id**, **dm_id**, **c_nid** - **TrackDelivery**

SCHEMA DIAGRAM

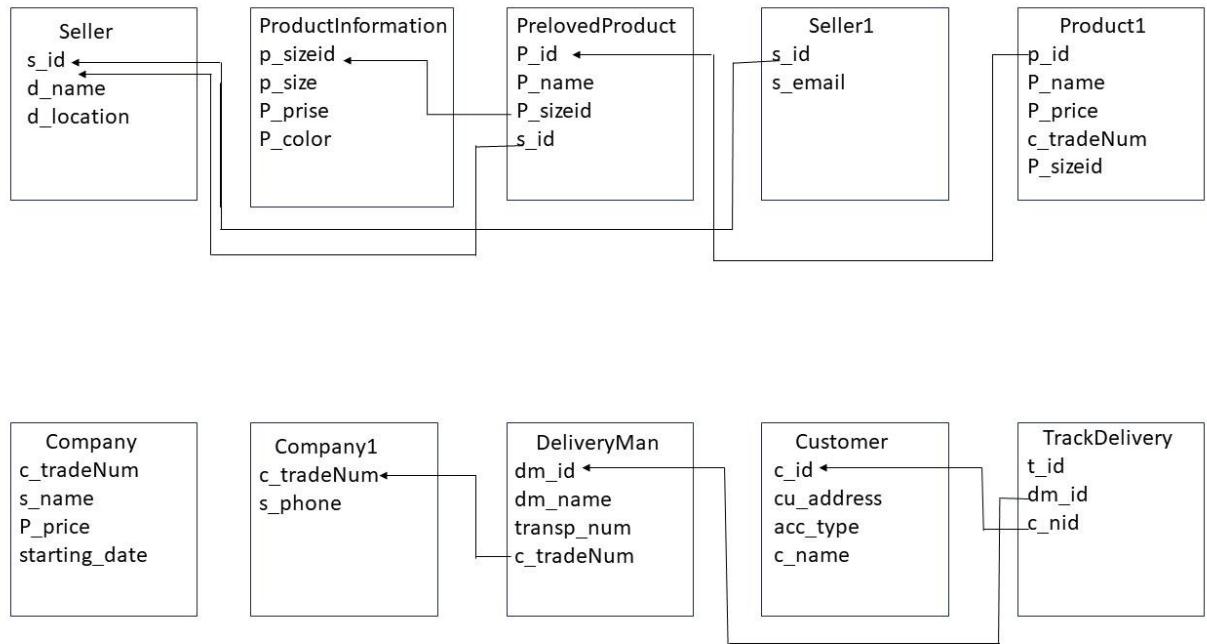


Table Creation

Create table Seller:

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

```
desc seller
desc Productinformation
desc prelovedproduct
desc seller1
desc product1
desc company
desc company1
desc deliveryman
desc customer
desc trackdelivery
```

The results pane shows the creation of the SELLER table:

Object Type	TABLE Object	SELLER							
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SELLER	S_ID	Number	-	-	0	1	-	-	-
	D_NAME	Varchar2	255	-	-	-	✓	-	-
	D_LOCATION	Varchar2	255	-	-	-	✓	-	-

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

Create table ProductInformation:

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

```
desc seller
desc Productinformation
desc prelovedproduct
desc seller1
desc product1
desc company
desc company1
desc deliveryman
desc customer
desc trackdelivery
```

The results pane shows the creation of the PRODUCTINFORMATION table:

Object Type	TABLE Object	PRODUCTINFORMATION							
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PRODUCTINFORMATION	P_SIZEID	Number	-	-	0	1	-	-	-
	P_SIZE	Varchar2	255	-	-	-	✓	-	-
	P_PRICE	Number	-	10	2	-	✓	-	-
	P_COLOR	Varchar2	255	-	-	-	✓	-	-

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

Create table prelovedproduct:

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the user has run the command `desc prelovedproduct`. Below the command, the table structure for `PRELOVEDPRODUCT` is displayed:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PRELOVEDPRODUCT	P_ID	Number	-	-	0	1	-	-	-
	P_NAME	Varchar2	255	-	-	-	✓	-	-
	P_SIZEID	Number	-	-	0	-	✓	-	-
	S_ID	Number	-	-	0	-	✓	-	-

At the bottom of the table structure, it says "1 - 4".

Create table seller1:

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the user has run the command `desc seller1`. Below the command, the table structure for `SELLER1` is displayed:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SELLER1	S_ID	Number	-	-	0	1	-	-	-
	S_EMAIL	Varchar2	255	-	-	-	✓	-	-

At the bottom of the table structure, it says "1 - 2".

Create table product1:

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the user has run the command 'desc seller' followed by 'desc product1'. The results show the structure of the PRODUCT1 table:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PRODUCT1	P_ID	Number	-	-	0	1	-	-	-
	P_NAME	Varchar2	255	-	-	-	✓	-	-
	P_PRICE	Number	-	10	2	-	✓	-	-
	C_TRADENUM	Number	-	-	0	-	✓	-	-
	P_SIZEID	Number	-	-	0	-	✓	-	-

Below the table definition, there is a note: '1 - 5'. The status bar at the bottom right indicates 'Application Express 2.1.0.0.39' and 'Copyright © 1999, 2006, Oracle. All rights reserved.'

Create table company:

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the user has run the command 'desc company'. The results show the structure of the COMPANY table:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
COMPANY	C_TRADENUM	Number	-	-	0	1	-	-	-
	S_NAME	Varchar2	255	-	-	-	✓	-	-
	P_PRICE	Number	-	10	2	-	✓	-	-
	STARTING_DATE	Date	7	-	-	-	✓	-	-

Below the table definition, there is a note: '1 - 4'. The status bar at the bottom right indicates 'Application Express 2.1.0.0.39' and 'Copyright © 1999, 2006, Oracle. All rights reserved.'

Create table company1:

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is entered:

```
desc seller
desc Productinformation
desc prelovedproduct
desc seller1
desc product1
desc company
desc company1
desc deliveryman
desc customer
desc trackdelivery
```

The results section displays the structure of the COMPANY1 table:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
COMPANY1	C_TRADENUM	Number	-	-	0	1	-	-	
	S_PHONE	Varchar2	20	-	-	-	✓	-	

Language: en-us Application Express 2.1 0 00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

Create table deliveryman:

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is entered:

```
desc seller
desc Productinformation
desc prelovedproduct
desc seller1
desc product1
desc company
desc company1
desc deliveryman
desc customer
desc trackdelivery
```

The results section displays the structure of the DELIVERYMAN table:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DELIVERYMAN	DM_ID	Number	-	-	0	1	-	-	
	DM_NAME	Varchar2	255	-	-	-	✓	-	
	TRANS_NUM	Varchar2	255	-	-	-	✓	-	
	C_TRADENUM	Number	-	-	0	-	✓	-	

Language: en-us Application Express 2.1 0 00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

Create table customer:

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is entered:

```
desc seller
desc Productinformation
desc prelovedproduct
desc seller1
desc product1
desc company
desc company1
desc deliveryman
desc customer
desc trackdelivery
```

The 'customer' line is highlighted in blue, indicating it is the current object being described. Below the code, the 'Object Type' is shown as 'TABLE Object CUSTOMER'. A table is displayed with the following columns:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER	C_NID	Number	-	-	0	1	-	-	-
	CU_ADDRESS	Varchar2	255	-	-	-	✓	-	-
	ACC_TYPE	Varchar2	50	-	-	-	✓	-	-
	C_NAME	Varchar2	255	-	-	-	✓	-	-

At the bottom of the interface, the status bar shows 'Language: en-us' and 'Copyright © 1999, 2006, Oracle. All rights reserved.' The system tray at the bottom right shows the date as 31/12/2023 and the time as 13:51.

Create table trackdelivery:

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is entered:

```
desc seller
desc Productinformation
desc prelovedproduct
desc seller1
desc product1
desc company
desc company1
desc deliveryman
desc customer
desc trackdelivery
```

The 'trackdelivery' line is highlighted in blue, indicating it is the current object being described. Below the code, the 'Object Type' is shown as 'TABLE Object TRACKDELIVERY'. A table is displayed with the following columns:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TRACKDELIVERY	T_ID	Number	-	-	0	1	-	-	-
	DM_ID	Number	-	-	0	-	✓	-	-
	C_NID	Number	-	-	0	-	✓	-	-

At the bottom of the interface, the status bar shows 'Language: en-us' and 'Copyright © 1999, 2006, Oracle. All rights reserved.' The system tray at the bottom right shows the date as 31/12/2023 and the time as 13:51.

Data Insertion

Data insertion in Seller table:

The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window displays the following SQL query:

```
select*from seller
select*from Productinformation
select*from prelovedproduct
select*from seller1
select*from product1
select*from company
select*from company1
select*from deliveryman
select*from customer
select*from trackdelivery
```

The Results tab shows the data inserted into the Seller table:

S_ID	D_NAME	D_LOCATION
2	Donald Trump	USA
3	Kamala Harris	USA
4	Rishi Sunak	UK
5	Georgia Meloni	Italy
6	Imran Khan	Pakistan
1	Joe Biden	USA

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

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

Data insertion in productinformation table:

The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window displays the following SQL query:

```
select*from seller
select*from Productinformation
select*from prelovedproduct
select*from seller1
select*from product1
select*from company
select*from company1
select*from deliveryman
select*from customer
select*from trackdelivery
```

The Results tab shows the data inserted into the Productinformation table:

P_SIZEID	P_SIZE	P_PRICE	P_COLOR
101	L	500	Black
102	M	200	Pink
103	S	100	White
104	XL	600	Red
105	XXL	800	Lavender

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

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

Data insertion in prelovedproduct table:

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL query is run:

```
select*from seller
select*from Productinformation
select*from prelovedproduct
select*from seller1
select*from product1
select*from company
select*from company1
select*from deliveryman
select*from customer
select*from trackdelivery
```

The results show a table with columns P_ID, P_NAME, P_SIZEID, and S_ID. The data is as follows:

P_ID	P_NAME	P_SIZEID	S_ID
11	Car	101	1
12	Home Theatre	102	2
13	Air Conditioner	103	3
14	Television	104	4
15	Mobile	105	5

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

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

Data insertion in Seller1 table:

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL query is run:

```
select*from seller
select*from Productinformation
select*from prelovedproduct
select*from seller1
select*from product1
select*from company
select*from company1
select*from deliveryman
select*from customer
select*from trackdelivery
```

The results show a table with columns S_ID and S_EMAIL. The data is as follows:

S_ID	S_EMAIL
1	emon@gmail.com
2	bjoy@gmail.com
3	siam@gmail.com
4	saf@gmail.com
5	ayon@gmail.com

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

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

Data insertion in product1 table:

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL query is run:

```
select*from seller
select*from Productinformation
select*from prelovedproduct
select*from seller1
select*from product1
select*from company
select*from company1
select*from deliveryman
select*from customer
select*from trackdelivery
```

The results show the following data inserted into the product1 table:

P_ID	P_NAME	P_PRICE	C_TRADENUM	P_SIZEID
11	Car	500	1001	101
12	Home Theatre	200	1002	102
13	Air Conditioner	100	1003	103
14	Television	600	1004	104
15	Mobile	800	1005	105

5 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.

Data insertion in company table:

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL query is run:

```
select*from seller
select*from Productinformation
select*from prelovedproduct
select*from seller1
select*from product1
select*from company
select*from company1
select*from deliveryman
select*from customer
select*from trackdelivery
```

The results show the following data inserted into the company table:

C_TRADENUM	S_NAME	P_PRICE	STARTING_DATE
1002	Bigoy	200	23-NOV-02
1003	Siam	100	12-MAY-09
1004	Saf	600	28-JUL-12
1005	Ayon	800	31-AUG-22
1001	Emon	500	05-JAN-01

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

Language: en-us Application Express 2.1 0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

Data insertion in company1 table:

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

```
select*from seller
select*from Productinformation
select*from prelovedproduct
select*from seller1
select*from product1
select*from company
select*from company1
select*from deliveryman
select*from customer
select*from trackdelivery
```

The results section displays a table with columns C_TRADENUM and S_PHONE, containing the following data:

C_TRADENUM	S_PHONE
1001	01825678901
1002	01987345607
1003	017651234560
1004	013127657909
1005	017345120987

5 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.

Data insertion in deliverman table:

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL query entered is identical to the one in the previous screenshot:

```
select*from seller
select*from Productinformation
select*from prelovedproduct
select*from seller1
select*from product1
select*from company
select*from company1
select*from deliveryman
select*from customer
select*from trackdelivery
```

The results section displays a table with columns DM_ID, DM_NAME, TRANSP_NUM, and C_TRADENUM, containing the following data:

DM_ID	DM_NAME	TRANSP_NUM	C_TRADENUM
505	Nayeem	YST	1001
506	Abid	UHL	1002
507	Zubayer	POI	1003
508	Bijoy	RLL	1004
509	Rakin	KPP	1005

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

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

Data insertion in customer table:

The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window displays the following SQL query:

```
select*from seller
select*from Productinformation
select*from prelovedproduct
select*from seller1
select*from product1
select*from company
select*from company1
select*from deliveryman
select*from customer
select*from trackdelivery
```

The Results tab shows the data inserted into the customer table:

C_NID	CU_ADDRESS	ACC_TYPE	C_NAME
29348	Uttara	Temporary	Anika
75689	Gulshan	Permanent	Suraya
67834	Narayanganj	Yearly	Mahira
78347	Narayanganj	Yearly	Tabassum
10023	Malibagh	Permanent	Ameena

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

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

Data insertion in trackdelivery table:

The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window displays the same SQL query as the previous screenshot:

```
select*from seller
select*from Productinformation
select*from prelovedproduct
select*from seller1
select*from product1
select*from company
select*from company1
select*from deliveryman
select*from customer
select*from trackdelivery
```

The Results tab shows the data inserted into the trackdelivery table:

T_ID	DM_ID	C_NID
100	505	10023
101	506	29348
102	507	75689
103	508	67834
104	509	78347

5 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.

QUERY WRITING

Query Writing (Write down the question and the answer. Give screenshot of the result of the query):

SINGLE ROW FUNCTIONS

- Change the name of delivery man in capital letter.

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is run:

```
SELECT seller.s_id, seller.d_name
FROM seller, prelovedproduct
WHERE seller.s_id(+)=prelovedproduct.s_id;

SELECT c_name, upper(c_name)
from customer;

SELECT dm_name, upper(dm_name)
From deliveryman;
```

The results show the names converted to uppercase:

DM_NAME	UPPER(DM_NAME)
Nayem	NAYEM
Abi	ABID
Zubayer	ZUBAYER
Bijoy	BIJOY
Rakin	RAKIN

5 rows returned in 0.01 seconds

- Find the length of customer's names.

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is run:

```
SELECT productinformation.p_sizeid, prelovedproduct.p_sizeid
FROM productinformation, prelovedproduct
WHERE productinformation.p_sizeid = prelovedproduct.p_sizeid;

SELECT seller.s_id, seller.d_name
FROM seller, prelovedproduct
WHERE seller.s_id(+)=prelovedproduct.s_id;

SELECT c_name, length(c_name)
from customer;
```

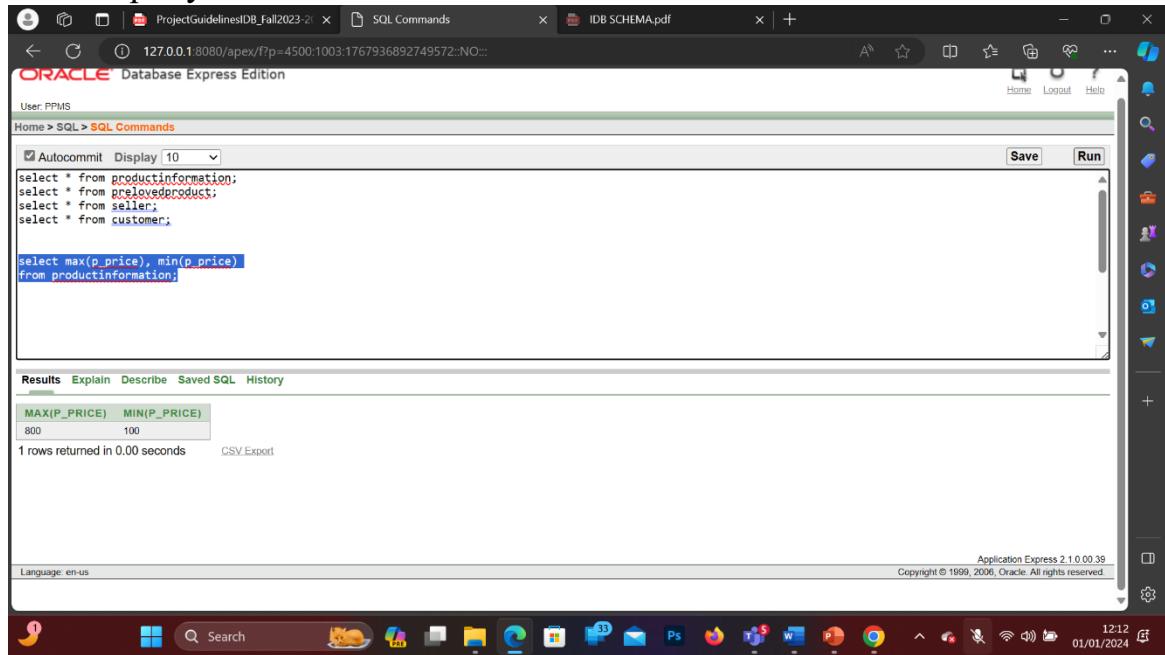
The results show the length of customer names:

C_NAME	LENGTH(C_NAME)
Anika	5
Suraya	7
Mahira	6
Tabassum	8
Amena	6

5 rows returned in 0.00 seconds

GROUP FUNCTIONS

- One customer wants to filter the products according to their prices. Now write a query based on this condition.



The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is entered:

```
select * from productinformation;
select * from prelovedproduct;
select * from seller;
select * from customer;

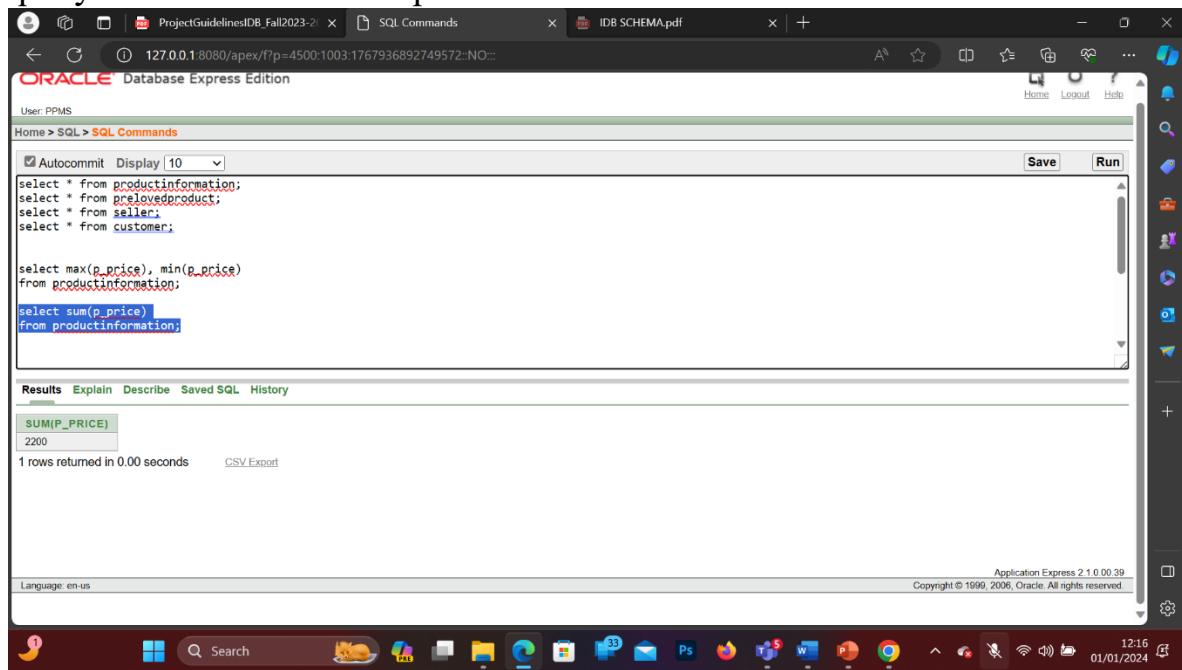
select max(p_price), min(p_price)
from productinformation;
```

The results section displays the output:

MAX(P_PRICE)	MIN(P_PRICE)
800	100

1 rows returned in 0.00 seconds

- One customer wants to buy all the products at the same time now write a query to calculate the total price for this customer.



The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is entered:

```
select * from productinformation;
select * from prelovedproduct;
select * from seller;
select * from customer;

select max(p_price), min(p_price)
from productinformation;

select sum(p_price)
from productinformation;
```

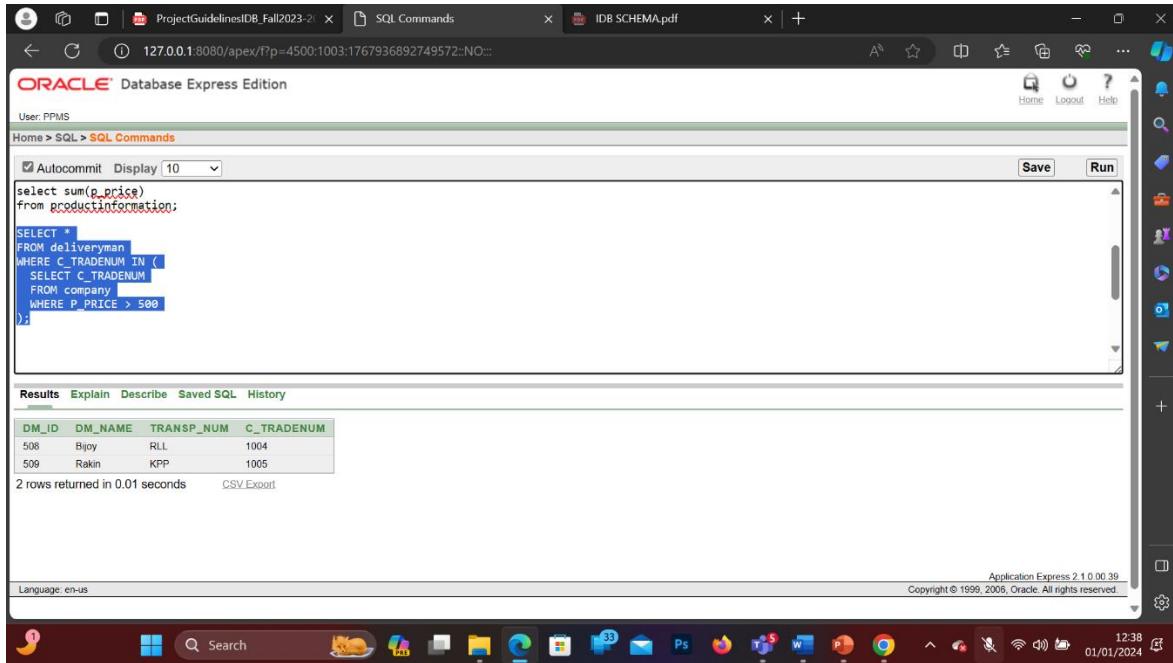
The results section displays the output:

SUM(P_PRICE)
2200

1 rows returned in 0.00 seconds

SUBQUERY

- Retrieve the information of deliverymen who have transported products associated with transactions from the "company" table where the product price is greater than 500. Include the deliveryman's ID, name, transported number, and the associated transaction number.



The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, a query is entered:

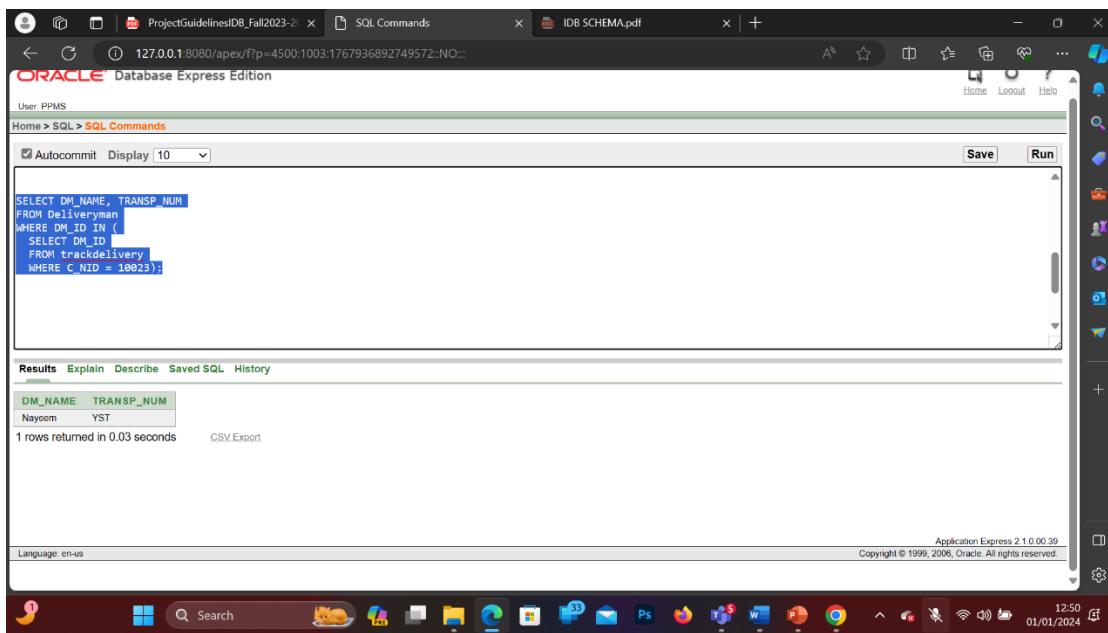
```
select sum(p.price)
from productinformation;
SELECT *
FROM deliveryman
WHERE C_TRADENUM IN (
    SELECT C_TRADENUM
    FROM company
    WHERE P_PRICE > 500
);
```

The results window displays the following data:

DM_ID	DM_NAME	TRANSP_NUM	C_TRADENUM
508	Bijoy	RLL	1004
509	Rakin	KPP	1005

2 rows returned in 0.01 seconds

- Retrieve the names and transportation numbers of deliverymen who are associated with the delivery tracking records in the "trackdelivery" table where the associated customer NID no is 10023.



The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, a query is entered:

```
SELECT DM_NAME, TRANSP_NUM
FROM Deliveryman
WHERE DM_ID IN (
    SELECT DM_ID
    FROM trackdelivery
    WHERE C_NID = 10023
);
```

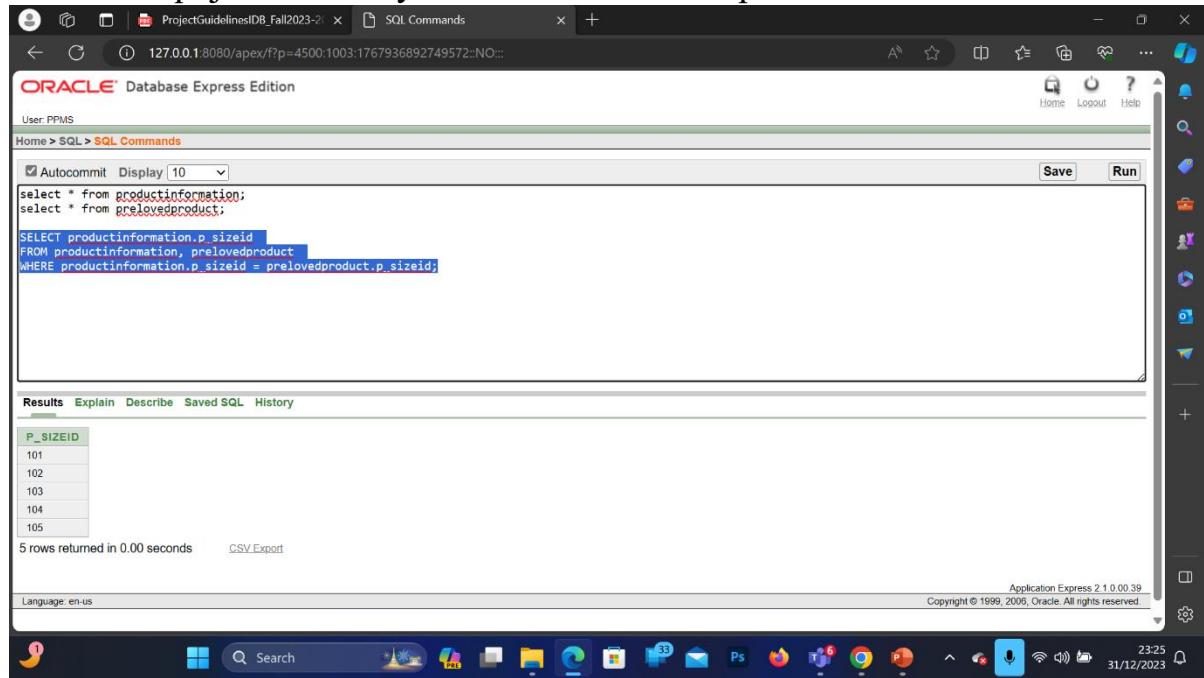
The results window displays the following data:

DM_NAME	TRANSP_NUM
Nayem	YST

1 rows returned in 0.03 seconds

JOINING

- Show an equijoin from any two tables where it's possible.



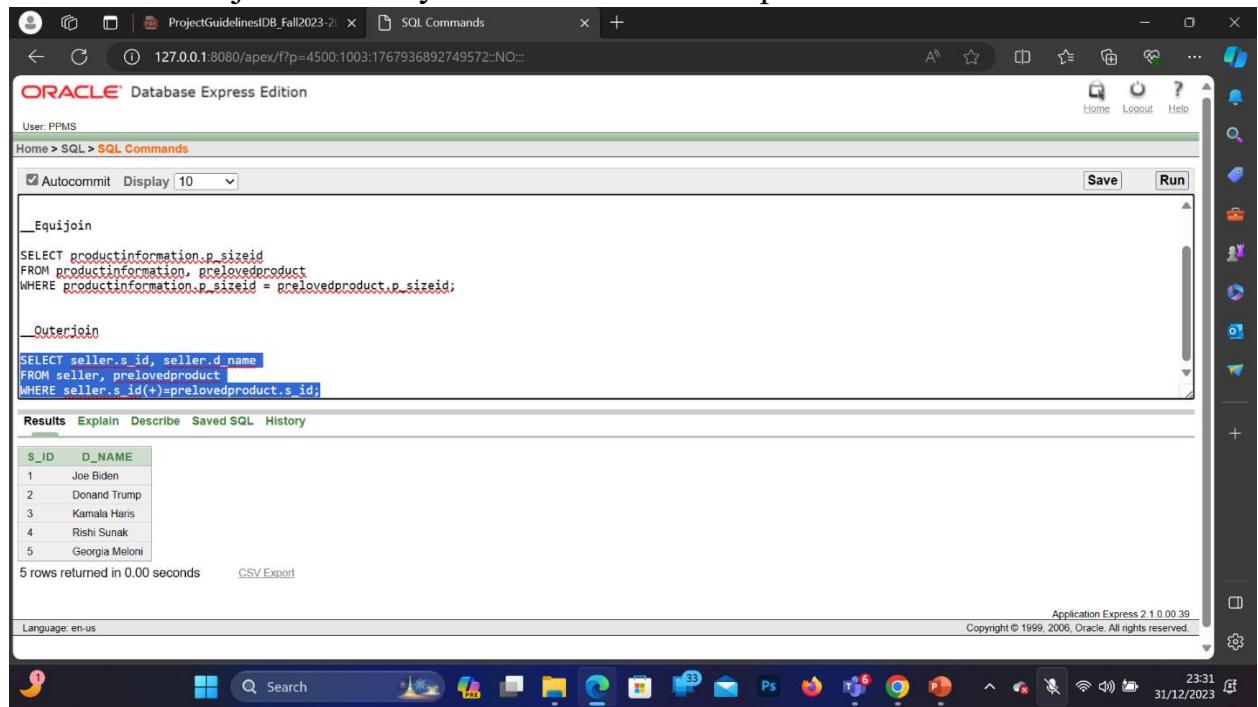
The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is entered:

```
select * from productinformation;
select * from prelovedproduct;

SELECT productinformation.p_sizeid
FROM productinformation, prelovedproduct
WHERE productinformation.p_sizeid = prelovedproduct.p_sizeid;
```

The results pane displays a table with one column, P_SIZEID, containing values 101, 102, 103, 104, and 105. The status bar at the bottom indicates "5 rows returned in 0.00 seconds". The application bar at the bottom right shows the date as 31/12/2023 and the time as 23:25.

- Show an outer join from any two tables where it's possible.



The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is entered:

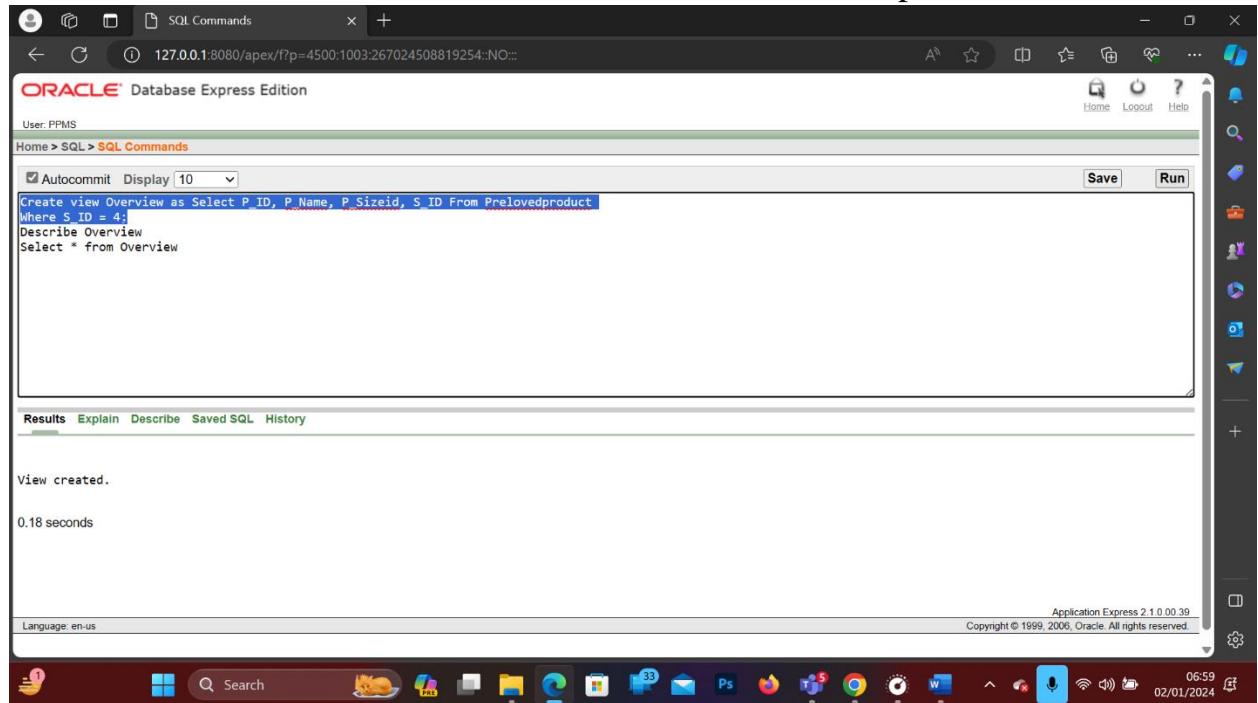
```
__Equijoin
SELECT productinformation.p_sizeid
FROM productinformation, prelovedproduct
WHERE productinformation.p_sizeid = prelovedproduct.p_sizeid;

__Outerjoin
SELECT seller.s_id, seller.d_name
FROM seller, prelovedproduct
WHERE seller.s_id(+)=prelovedproduct.s_id;
```

The results pane displays a table with two columns, S_ID and D_NAME, containing data for five rows. The application bar at the bottom shows the date as 31/12/2023 and the time as 23:31.

VIEW

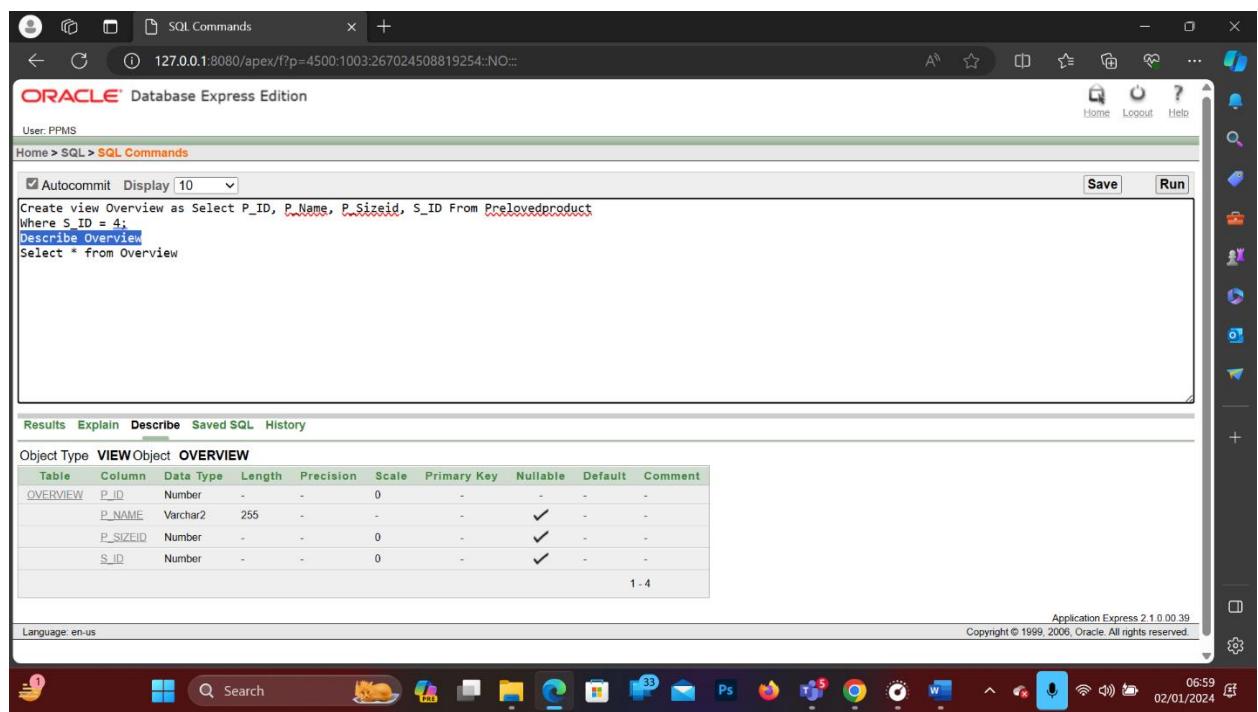
- Create a view named as “Overview” where Seller id 4 will be shown over the columns P_ID, P_Name, P_Sizeid, S_ID from Prelovedproduct table?



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

```
Create view Overview as Select P_ID, P_Name, P_Sizeid, S_ID From Prelovedproduct
Where S_ID = 4;
Describe Overview
Select * from Overview
```

The results pane shows the message "View created." and a execution time of "0.18 seconds". The status bar at the bottom right indicates "Application Express 2.1 0.00.39" and "Copyright © 1999, 2006, Oracle. All rights reserved".



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

```
Create view Overview as Select P_ID, P_Name, P_Sizeid, S_ID From Prelovedproduct
Where S_ID = 4;
Describe Overview
Select * from Overview
```

The results pane shows the message "Object Type: VIEW Object: OVERVIEW" and a table of columns:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
OVERVIEW	P_ID	Number	-	-	0	-	-	-	
	P_NAME	Varchar2	255	-	-	✓	-	-	
	P_SIZEID	Number	-	-	0	✓	-	-	
	S_ID	Number	-	-	0	✓	-	-	

The status bar at the bottom right indicates "Application Express 2.1 0.00.39" and "Copyright © 1999, 2006, Oracle. All rights reserved".

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The user is executing a SQL script to create a view named 'Overview'.

```

CREATE VIEW Overview AS
Select P_ID, P_Name, P_Sizeid, S_ID From Prelovedproduct
Where S_ID = 4;
Describe Overview
Select * from Overview

```

The results of the query are displayed in a table:

P_ID	P_Name	P_Sizeid	S_ID
14	Television	104	4

1 rows returned in 0.03 seconds

CSV Export

Application Express 2.1 0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

- Create a view name “Overview2” to show the joining between the tables Seller and Prelovedproduct.

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The user is creating a new view named 'Overview2'.

```

CREATE VIEW Overview2 AS
SELECT S.S_ID AS Seller_ID, S.D_Name, P.P_Name, P.P_Sizeid, P.S_ID AS Product_ID
FROM Seller S
LEFT JOIN Prelovedproduct P ON P.S_ID = S.S_ID;

Describe Overview2
Select * from Overview2

```

The results of the query are displayed in a table:

View created.

0.03 seconds

Application Express 2.1 0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

User: PPMS

Home > SQL > SQL Commands

Autocommit

```
CREATE VIEW Overview2 AS
SELECT S.S_ID AS Seller_ID, S.D_Name, P.P_Name, P.P_Sizeid, P.S_ID AS Product_ID
FROM Seller S
LEFT JOIN Prelovedproduct P ON P.S_ID = S.S_ID;

Describe Overview2
Select * from Overview2
```

Results Explain Describe Saved SQL History

Object Type: VIEW Object: OVERVIEW2

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
OVERVIEW2	SELLER_ID	Number	-	-	0	-	-	-	-
	D_NAME	Varchar2	255	-	-	-	✓	-	-
	P_NAME	Varchar2	255	-	-	-	✓	-	-
	P_SIZEID	Number	-	-	0	-	✓	-	-
	PRODUCT_ID	Number	-	-	0	-	✓	-	-

1 - 5

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

07:07 02/01/2024

User: PPMS

Home > SQL > SQL Commands

Autocommit

```
CREATE VIEW Overview2 AS
SELECT S.S_ID AS Seller_ID, S.D_Name, P.P_Name, P.P_Sizeid, P.S_ID AS Product_ID
FROM Seller S
LEFT JOIN Prelovedproduct P ON P.S_ID = S.S_ID;

Describe Overview2
Select * from Overview2
```

Results Explain Describe Saved SQL History

SELLER_ID	D_NAME	P_NAME	P_SIZEID	PRODUCT_ID
1	Joe Biden	Car	101	1
2	Donald Trump	Home Theatre	102	2
3	Kamala Harris	Air Conditioner	103	3
4	Rishi Sunak	Television	104	4
5	Georgia Meloni	Mobile	105	5
6	Imran Khan	-	-	-

6 rows returned in 0.03 seconds [CSV Export](#)

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

07:07 02/01/2024

RELATIONAL ALGEBRA

- Find the name of the customer number where customer name is tabasssum.
➤ $\Pi_{c_nid}(\sigma_{c_name= "Tabassum"} (Number))$
- Find the product price where amount is greater than 100 dollars.
➤ $\Pi_{p_price}(\sigma_{amount > 100} (Price))$
- Find the location of seller where seller name is Imran Khan.
➤ $\Pi_{d_location}(\sigma_{d_name= "Imran Khan"} (Customer's\ name))$
- Find the name of the customer whose ID number is 10023.
➤ $\Pi_{c_name}(\sigma_{c_nid= "10023"} (Customer))$
- Find the name of delivery man whose transport number is RLL.
➤ $\Pi_{dm_name}(\sigma_{Transp_num= "RLL"} (Deliveryman's\ name))$

DB CONNECTION

1. Mysql java connector (jar file)
You can go to “**mysql java connector maven**” any version you like.
2. You need to search (xampp apache mariadb perl php) install xampp server normally.
3. Select the xampp icon control and then start both apache and mysql. Next go to mysql admin panel.
4. Now create a Database, inside of it create table and insert the values in the table.
The table in mysql is:

s_id	d_name	d_location
2	Donand Trump	USA
3	Kamala Haris	USA
4	Rishi Sunak	UK
5	Georgia Meloni	Italty

Now we need an IDE.

Before that, we need to understand the steps of DB connection.

- Register Driver

```
Class.forName( className: "com.mysql.jdbc.Driver");
```

- Connect the DB

```
Connection con = DriverManager.getConnection(  
    url: "jdbc:mysql://localhost:3306/PPMS_Project", user: "root", password: "");
```

- Create statement:

```
Statement stmt = con.createStatement();
```

- Execute the query () in the resultstmtnt

```
ResultSet rs = stmt.executeQuery( sql: "select * from seller");
```

The while loop iterates through the rows of the result set (rs). The next() method advances the cursor to the next row in the result set and returns true if there is a next row, and false if there are no more rows.

```
while (rs.next())  
    System.out.println(rs.getInt( columnIndex: 1 ) + " " + rs.getString( columnIndex: 2 )  
        + " " + rs.getString( columnIndex: 3 ));
```

- Connection close ()

```
con.close();
```

5. Now write the DB connection code and add the jar file location in the libraries. The output will be

```
2 Donand Trump USA  
3 Kamala Haris USA  
4 Rishi Sunak UK  
5 Georgia Meloni Italty
```

CONCLUSION:

Our project on preloved product management is built so that it can be used as a database system of management of preloved products in various sectors. We try to incorporate every kind of entity and to store its data in the most straightforward manner imaginable. Through this project, we used the theoretical knowledge we gained in this Introduction to Database course for a real-world project. Hopefully, the information we have gained from this project and the training will be useful when developing any database-related plans we may take on in the future. Additionally, we are eager to work on this project and are prepared to add new features.