



AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

Department of the Faculty of Science and Technology

Summer 2024-2025

Project Report On

Laundry Service Management System

Supervised By

JUENA AHMED NOSHIN

Name	ID	Contribution
Tanvir Arabi	22-47857-2	Design user interface, Normalization, Query implementation, and Data insertion (34%)
Bushra Kaiser	22-47679-2	Scenario Description, ER Diagram, Schema Diagram, Data Insertion, and Query Implementation – (33%)
MD. Anas Khan	20-43091-1	Design user interface, Normalization, Query implementation, and Data insertion (33%)
MD Ridoyuzzaman	22-47287-1	0%

ADVANCED DATABASE MANAGEMENT SYSTEM

Section: B

Group No: 04

Contents:

Content Name	Page No
Introduction	3
Project Proposal	3-4
User Interface Planning	4-9
Scenario Description	9-10
ER Diagram	10
Normalization	11-12
Schema Diagram	13
Table Creation	13-17
Data Insertion	18-22
Basic PL/SQL	22-27
Advance PL/SQL	28-37
Conclusion	37

Title: Laundry Service Management System.

Introduction:

The goal of our project is to simplify and organize the management of a laundry operation. It facilitates the laundry service's management of employees, customers, and services provided. Regular customers will obtain a unique membership card with discounted prices and they seek laundry delivery and pickup straight from their homes. Lastly, the system manages utility bills, which are paid for by clients. It is a tool that simplifies the whole laundry service process for the company and its customers and makes life easier.

Project Proposal :

(i) Problem Background:

The laundry marketplace has a lot of problems, particularly for small and medium-sized companies. These problems impact both the service providers and their clients. Managing staff, client orders, and invoicing by hand and in the old-fashioned way typically leads to errors, inefficiencies, and bad customer service. In today's fast-paced world, when clients want speedy and dependable service, this dilemma is much worse. For students in higher education and those who are employed, doing laundry may be a pain since they are typically busy and don't have a lot of time. They could have trouble making time to go to the laundromat or do laundry at home. Also, laundry services typically don't meet their demands, including providing home pickups or flexible hours, which makes it tougher for people to trust these services.

(ii) Objective:

The Laundry Service Management System solves these problems by automating important operations like managing employees, helping customers, and invoicing, which makes the experience better for both the business and the client. The system has features that save time for students in higher education and those who are employed, such as the ability to request home service, get membership perks, and pay bills and make payments easily.

(iii) Benefits:

The Laundry Service Management System increases productivity by automating processes, including processing payments, coordinating staff, and responding to client demands. It offers a practical solution with home pickup and delivery services, saving valuable time for jobholders and bachelor students. Direct utility bill payment, membership card discounts, and laundry

status monitoring are all available to customers. Along with increasing staff productivity and ensuring timely and adaptable services, the technology also helps reduce operating expenses. Technology makes operations easier and gives clients a more convenient experience with its consolidated data, real-time tracking, and scalability.

User Interface planning :

(Registration & Login)

The image displays two wireframe designs for a customer registration form, side-by-side. Both designs feature a blue header bar with the 'Laundry Service' logo and the text 'Customer Registration' and 'Register for our premium laundry services'. The left design shows a simplified form with fields for 'Full Name', 'Phone Number', 'Email Address', and 'Address', each with an input field and placeholder text. A large blue button at the bottom contains the text 'Register Now' with a user icon. Below the button is a small note: 'By registering, you agree to our [Terms of Service](#)'. The right design is more detailed, showing a 'Full Name' field with a placeholder 'Enter your full name', a 'Phone Number' field with a placeholder 'Enter your phone number', an 'Email Address' field with a placeholder 'Enter your email', and an 'Address' field with a placeholder 'Your complete address'. It includes a 'Customer Registration' title and a 'Register for our premium laundry services' subtitle. A 'Manage addresses...' link is visible next to the email field. The right design also features a 'Tanvir Arabi' placeholder entry in the phone number field, indicating a previous entry or suggestion. It includes password fields for 'Password' and 'Confirm Password' with placeholder text 'Create a password' and 'Confirm your password' respectively. Both designs include a large blue 'Register Now' button at the bottom and a note below it: 'By registering, you agree to our [Terms of Service](#)'.

 Laundry Service
Customer Registration
 Register for our premium laundry services

Full Name

Phone Number

Email Address

01784193946
+880 178 419 394 6

Address

Password

Confirm Password

Register Now

By registering, you agree to our [Terms of Service](#)

<div style="position: absolute; left

Laundry Service
Management Dashboard

JS John Smith
Administrator [Logout](#)

24 Today's Orders	8 In Progress
12 Ready for Pickup	4 Delivered

[Customers](#) [Orders](#) [Employees](#)

Search customers... [+](#)

Customer List

Name	Phone	Orders	Actions
Tanvir Arabi	(555) 123-4567	12	Edit Delete
Bushra Kaiser	(555) 987-6543	8	Edit Delete
Anas Khan	(555) 456-7890	5	Edit Delete
Tanvir	(555) 234-5678	15	Edit Delete

Laundry Service
Welcome Back!
[Login to access your account](#)

Welcome to our laundry service! We're glad to have you back.

[Phone](#) [Email](#)

Phone Number
 Enter your phone number

Password
 Enter your password [?](#)

Remember me [Forgot Password?](#)

Login

Don't have an account? [Register now](#)

Laundry Service
Welcome Back!
[Login to access your account](#)

Welcome to our laundry service! We're glad to have you back.

[Phone](#) [Email](#)

Email Address
 Enter your email

Password
 Enter your password [?](#)

Remember me [Forgot Password?](#)

Login

Don't have an account? [Register now](#)

Laundry Service

Welcome Back!

Login to access your account

Welcome to our laundry service! We're glad to have you back.

Email Address

Password

Remember me

22-47857-2@student.aiub.edu
tanvirarabi8@gmail.com
22-47679-2@student.aiub.edu
tanvirarabi9@gmail.com
tanvirarabi0@gmail.com
tanvirarabi01@gmail.com

Don't have an account? [Register now](#)

Laundry Service

Welcome Back!

Login to access your account

Welcome to our laundry service! We're glad to have you back.

Email Address

Password

Remember me [Forgot Password?](#)

Login

Don't have an account? [Register now](#)

Dashboard:

Laundry Service

Dashboard

Welcome back, Sarah! Ready to get your laundry done?

Select Gender Category

Male Female

Available Services

Category	Description	Quantity	Price
Regular Clothing			
Shirts	50 TK per item	<input type="button" value="-"/> <input type="button" value="0"/> <input type="button" value="+"/>	
Pants	60 TK per item	<input type="button" value="-"/> <input type="button" value="0"/> <input type="button" value="+"/>	
Panjabi/Pajama	100 TK per item	<input type="button" value="-"/> <input type="button" value="0"/> <input type="button" value="+"/>	

Your Selection

No items selected yet

Total: 0 TK

Submit to Clean

Laundry Service

Dashboard

Welcome back, Sarah! Ready to get your laundry done?

Select Gender Category

Male Female

Available Services

Category	Description	Quantity	Price
Regular Clothing			
Shirts	50 TK per item	<input type="button" value="-"/> <input type="button" value="1"/> <input type="button" value="+"/>	50 TK
Pants	60 TK per item	<input type="button" value="-"/> <input type="button" value="1"/> <input type="button" value="+"/>	60 TK
Panjabi/Pajama	100 TK per item	<input type="button" value="-"/> <input type="button" value="1"/> <input type="button" value="+"/>	100 TK

Your Selection

Item	Quantity	Price
Shirts x1	1	50 TK
Pants x1	1	60 TK
Panjabi/Pajama x1	1	100 TK
Total:		210 TK

Submit to Clean

Schedule & Payment option

Laundry Service
 Quick order form

Dates

Received	Delivery
<input type="text" value="08/22/2025"/>	<input type="text" value="08/23/2025"/>

Service

 Pick & Drop	 Self Service
-----------------	------------------

Payment

 bKash	 Rocket	 Nagad
 Card	 Cash	

Summary

Service:	₹ 300
Delivery:	₹ 60
Tax:	₹ 30
Total:	₹ 390

[Pay Now](#)

Laundry Service
 Quick order form

Dates

Received	Delivery
<input type="text" value="08/22/2025"/>	<input type="text" value="08/23/2025"/>

Service

 Pick & Drop	 Self Service
-----------------	------------------

Payment

 bKash	 Rocket	 Nagad
 Card	 Cash	

Summary

Service:	₹ 300
Delivery:	₹ 0
Tax:	₹ 30
Total:	₹ 330

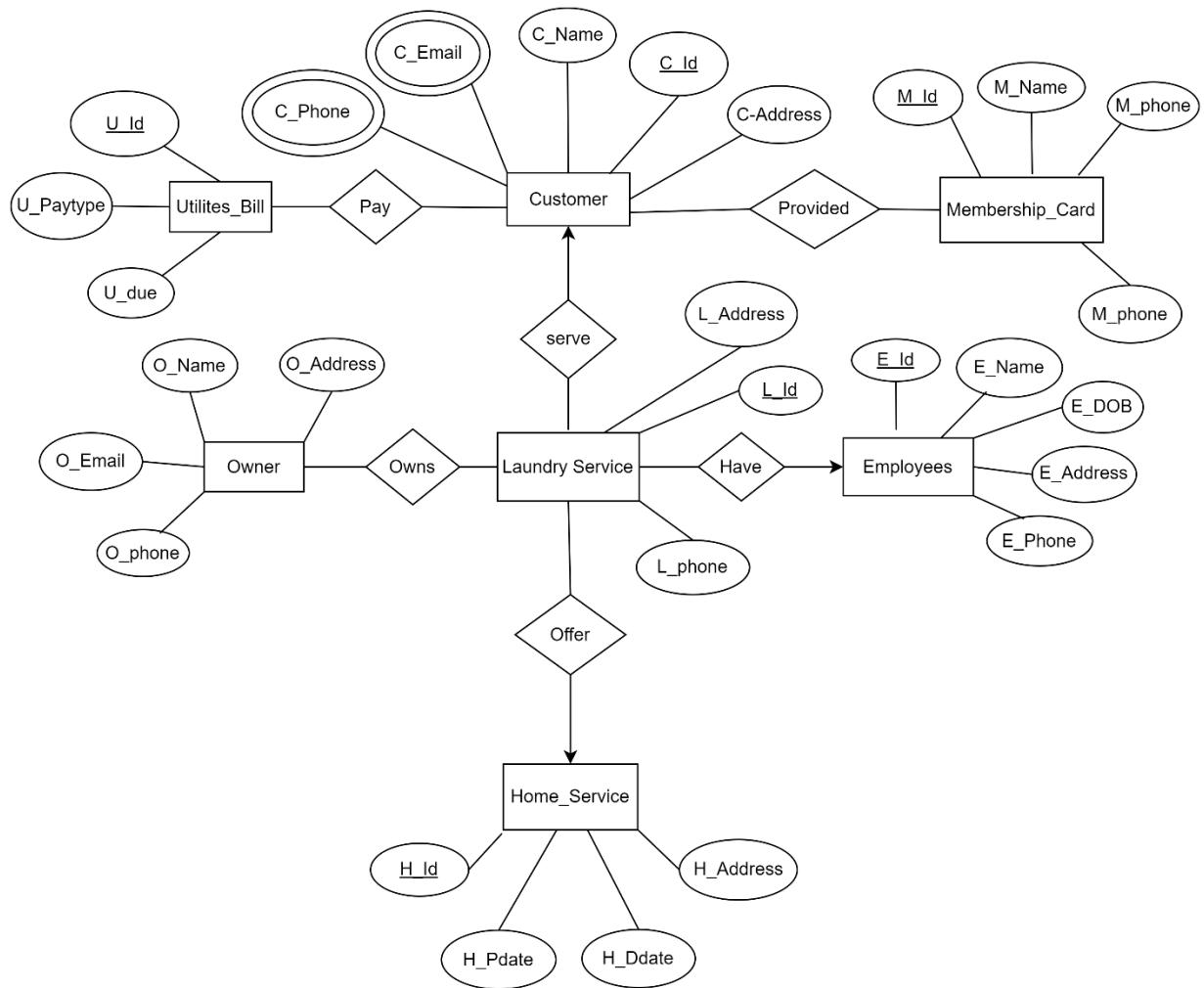
[Pay Now](#)

Scenario Description :

In the Laundry Service Management System, a Laundry Service can have many employees, but each employee works for only one Laundry Service. Each Employee is identified by a unique EmployeeID and has attributes such as name, age, address, date of birth, and phone number. A Laundry Service is identified by a unique LaundryServiceID and is characterized by its address and contact phone number. A Laundry Service may be owned by exactly one owner, and similarly, each Owner rents exactly one Laundry Service. The Owner is identified by their name, address, email, and phone number. A Laundry Service may serve many Customers, where each Customer is given a unique customer ID along with details like name, address, email, and phone number. A Customer can have multiple email addresses and phone numbers for communication purposes. For regular Customers, a special membership card will be provided. This card will include a MemberCardID, member name, discount, and phone number. The customer is responsible for paying a Utility Bill, which has a unique UtilityBillID, along with the payment type and the amount due for the services rendered. A Laundry Service may offer Home Services, where an Employee picks up laundry from the Customer's Home Address and delivers it back after cleaning. Each Home Service is identified by a unique HomeServiceID. Home Service Records store the details

such as the HomeServiceID, pickup date, delivery date, employee assigned to the service, and the Customer's Home Address. Each Home Service can have one or more Employees assigned to it, depending on the nature of the service. Employees are responsible for picking up and delivering laundry. Customers can request Home Service through the system, and it will be tracked for payment and service completion.

ER Diagram



Normalization:

1NF (First Normal Form)

A table is in 1NF if it meets these criteria:

- Each column contains a single, atomic value.
- There are no repeating groups of columns.

The ER diagram suggests the following initial tables. Primary keys are bolded.

- Customer (**C_Id**, C_Name, C_Address, C_Email, C_Phone)
- Utilities_Bill (**U_Id**, U_paytype, U_due, Pay)
- Membership_Card (**M_Id**, M_Name, M_phone, Provided)
- Owner (**O_Id**, O_Name, O_Address, O_Email, O_phone, Owns)
- Employees (**E_Id**, E_Name, E_DOB, E_Address, E_phone, Have)
- Laundry_Service (**L_Id**, L_Address, L_phone, Serve)
- Home_Service (**H_Id**, H_Pdate, H_Ddate, H_Address, Offer)

2NF (Second Normal Form)

A table is in 2NF if it is in 1NF and all non-key attributes are fully dependent on the primary key. This is only relevant for tables with a composite primary key.

Based on the ER diagram, we need to create linking tables to represent the relationships and ensure full dependency.

- Customer (C_Id, C_Name, C_Address, C_Email, C_Phone) - C_Id is the primary key. No changes needed.
- Utilities Bill (U_Id, U_paytype, U_due) - We remove Pay as it's a relationship. U_Id is the primary key.
- Membership Card (M_Id, M_Name, M_phone) - We remove Provided as it's a relationship. M_Id is the primary key.
- Owner (O_Id, O_Name, O_Address, O_Email, O_phone) - We remove Owns as it's a relationship. O_Id is the primary key.
- Employees (E_Id, E_Name, E_DOB, E_Address, E_phone) - We remove Have as it's a relationship. E_Id is the primary key.
- Laundry Service (L_Id, L_Address, L_phone) - We remove Serve as it's a relationship. L_Id is the primary key.
- Home Service (H_Id, H_Pdate, H_Ddate, H_Address) - We remove Offer as it's a relationship. H_Id is the primary key.

Now, we create the new linking tables for the relationships:

- Customer_Pays_Bill (C_Id, U_Id) - Composite primary key (C_Id, U_Id). Represents the Pay relationship.
- Customer_Has_Membership (C_Id, M_Id) - Composite primary key (C_Id, M_Id). Represents the Provided relationship.
- Owner_Owns_Laundry (O_Id, L_Id) - Composite primary key (O_Id, L_Id). Represents the Owns relationship.
- Laundry_Serves_Customer (L_Id, C_Id) - Composite primary key (L_Id, C_Id). Represents the Serve relationship.
- Employees_Have_Laundry (E_Id, L_Id) - Composite primary key (E_Id, L_Id). Represents the Have relationship.
- Laundry_Offers_Home_Service (L_Id, H_Id) - Composite primary key (L_Id, H_Id). Represents the Offer relationship.

All non-key attributes are now fully dependent on the primary key, satisfying 2NF.

3NF (Third Normal Form)

A table is in 3NF if it is in 2NF and has no transitive dependencies. A transitive dependency is when a non-key attribute depends on another non-key attribute.

Reviewing the tables, there don't appear to be any transitive dependencies. All attributes directly relate to their respective primary keys. For instance, in the Customer table, C_Name, C_Address, C_Email, and C_Phone all directly describe a C_Id. The same is true for all other tables. Therefore, the tables from 2NF are already in 3NF.

Summary of Table Names After Normalization:

- **Customer:** Customer
- **Membership_Card:** Membership_Card
- **Owner:** Owner
- **Employees:** Employees
- **Laundry_Service:** Laundry_Service
- **Home_Service:** Home_Service
- **Utilites_Bill:** Utilites_Bill
- **Customer_Pays_Bill:** Cust_Pays_Bill
- **Laundry_Service_Offers_Home_Service:** LS_Offers_HS

Schema Diagram:



Table creation using SQL:

1.Customer:

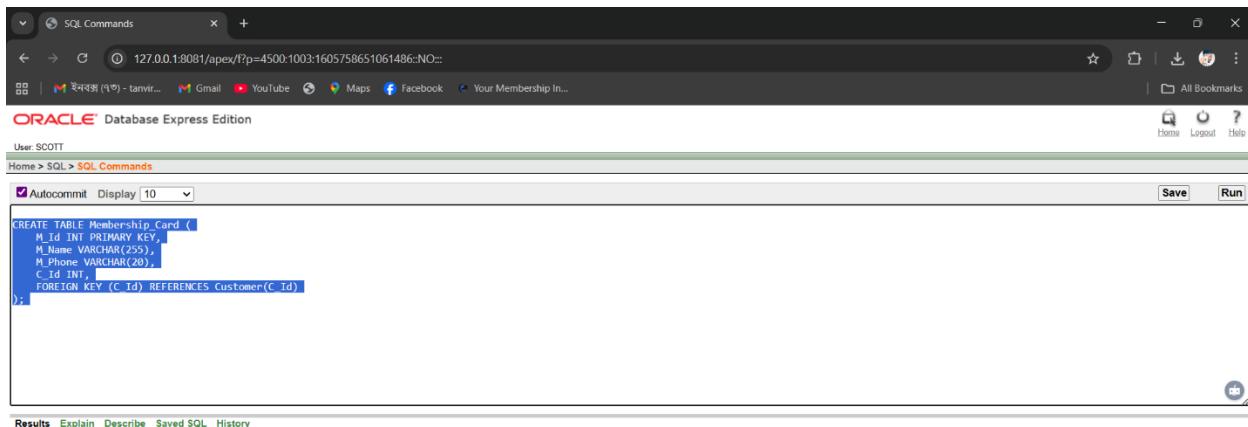
```

CREATE TABLE Customer (
    C_Id INT PRIMARY KEY,
    C_Name VARCHAR(255),
    C_Address VARCHAR(255),
    C_Email VARCHAR(255),
    C_Phone VARCHAR(20)
);
    
```

Table created.
0.11 seconds

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

2. Membership card

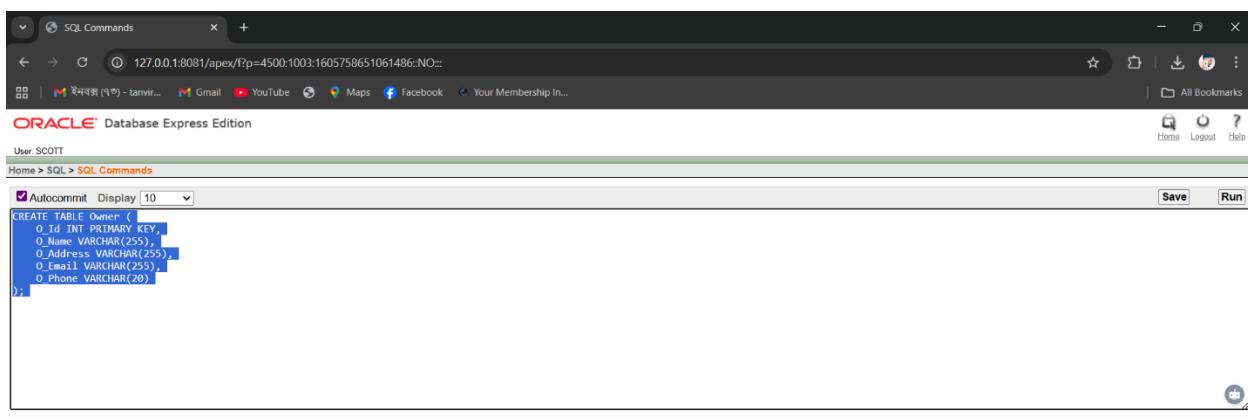


CREATE TABLE Membership_Card (
 M_Id INT PRIMARY KEY,
 M_Name VARCHAR(255),
 M_Phone VARCHAR(20),
 C_Id INT,
 FOREIGN KEY (C_Id) REFERENCES Customer(C_Id)
);

Table created.
0.01 seconds

Language: en-gb Application Express 2.1 0.00.39
Copyright © 1995, 2005, Oracle. All rights reserved.

3. Owner

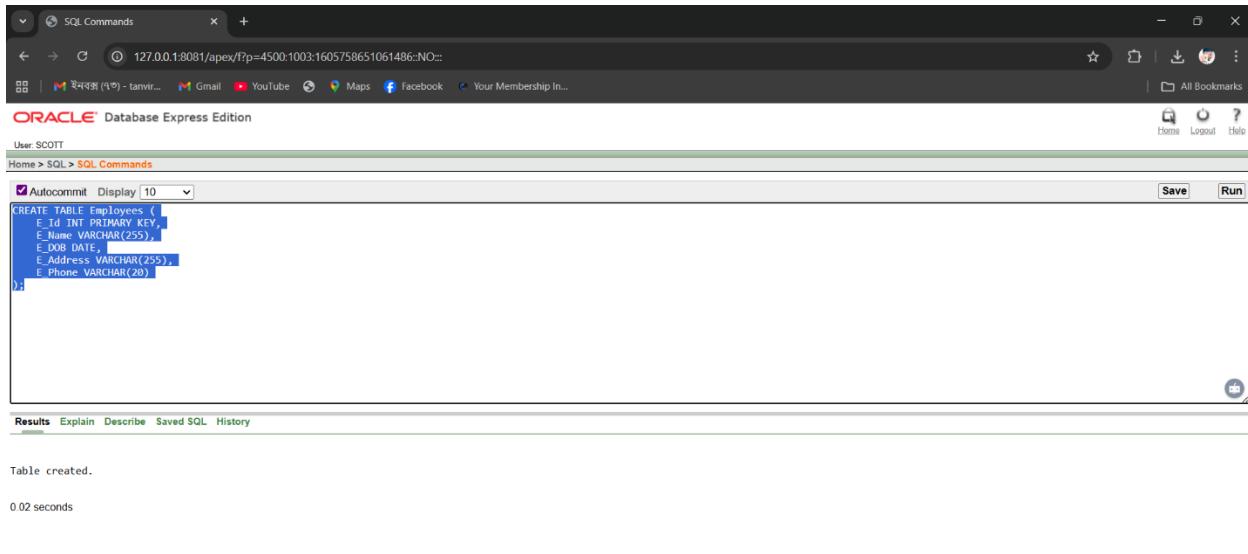


CREATE TABLE Owner (
 O_Id INT PRIMARY KEY,
 O_Name VARCHAR(255),
 O_Address VARCHAR(255),
 O_Email VARCHAR(255),
 O_Phone VARCHAR(20)
);

Table created.
0.01 seconds

Language: en-gb Application Express 2.1 0.00.39
Copyright © 1995, 2005, Oracle. All rights reserved.

4.Employees



SQL Commands

127.0.0.1:8081/apex/f?p=4500:1003:1605758651061486::NO::

User SCOTT

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
CREATE TABLE Employees (
    E_Id INT PRIMARY KEY,
    E_Name VARCHAR(255),
    E_Dob DATE,
    E_Address VARCHAR(255),
    E_Phone VARCHAR(20)
);
```

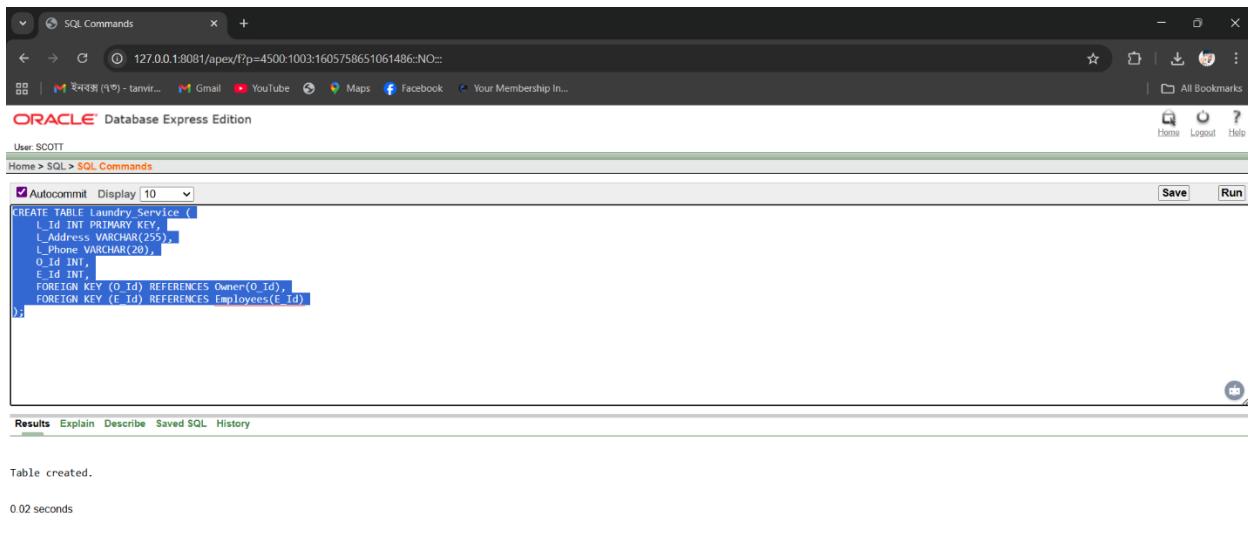
Results Explain Describe Saved SQL History

Table created.

0.02 seconds



5.Laundry Service



SQL Commands

127.0.0.1:8081/apex/f?p=4500:1003:1605758651061486::NO::

User SCOTT

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
CREATE TABLE Laundry_Service (
    L_Id INT PRIMARY KEY,
    L_Address VARCHAR(255),
    L_Phone VARCHAR(20),
    O_Id INT,
    E_Id INT,
    FOREIGN KEY (O_Id) REFERENCES Owner(O_Id),
    FOREIGN KEY (E_Id) REFERENCES Employees(E_Id)
);
```

Results Explain Describe Saved SQL History

Table created.

0.02 seconds



6. Home Service

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The URL in the address bar is 127.0.0.1:8081/apex/?p=4500:1003:1605758651061486:NO... The page title is "SQL Commands". The top navigation bar includes links for Home, Gmail, YouTube, Maps, Facebook, and Your Membership. A star icon for bookmarks is also present. The main content area displays the following SQL code:

```
CREATE TABLE Home_Service (
    H_Id INT PRIMARY KEY,
    H_Service VARCHAR(255),
    H_Date DATE,
    H_DDate DATE
);
```

Below the code, there are "Save" and "Run" buttons. At the bottom, there are tabs for Results, Explain, Describe, Saved SQL, and History.

Table created.

0.01 seconds



7. Utilities Bill

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The URL in the address bar is 127.0.0.1:8081/apex/f?p=4500:1003:1605758651061486::NO:::. The page title is "SQL Commands". The top navigation bar includes links for Home, SQL, and Help. The main content area shows a SQL command for creating a table:

```
CREATE TABLE Utilities_Bill (
    U_Id INT PRIMARY KEY,
    U_Paytype VARCHAR(50),
    U_DueAmount DECIMAL(10, 2), -- Use DECIMAL for currency
    L_Id INT,
    FOREIGN KEY (L_Id) REFERENCES Laundry_Service(L_Id)
);
```

The "Autocommit" checkbox is checked. On the right side of the editor, there are "Save" and "Run" buttons. Below the editor, there are tabs for Results, Explain, Describe, Saved SQL, and History.

Table created.

0.01 seconds



8.Cust Pays Bill

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

```
CREATE TABLE Cust Pays Bill (
    C_Id INT,
    U_Id INT,
    PRIMARY KEY (C_Id, U_Id),
    FOREIGN KEY (C_Id) REFERENCES Customer(C_Id),
    FOREIGN KEY (U_Id) REFERENCES Utilities_Bill(U_Id)
);
```

Below the code, the message "Table created." is displayed, followed by "0.00 seconds".



9.Offer Hs

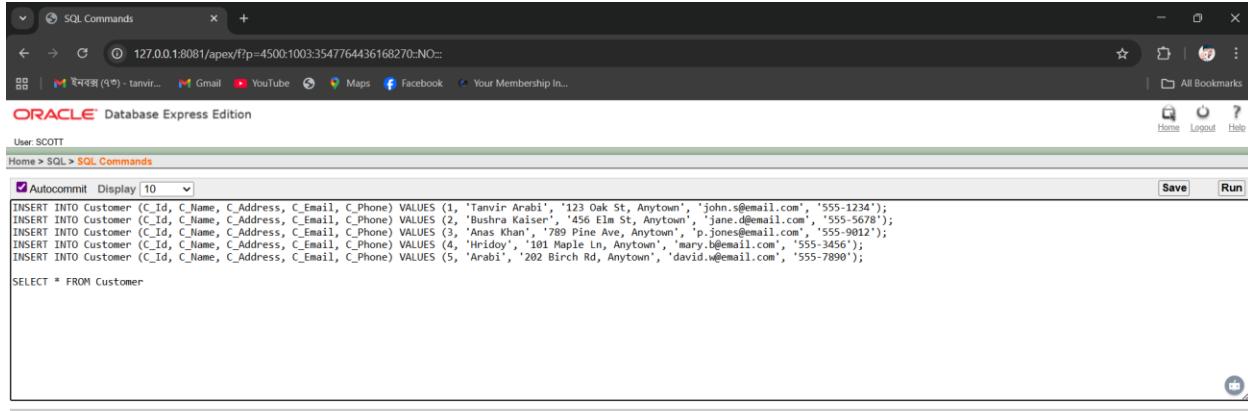
The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is entered:

```
CREATE TABLE LS Offers HS (
    L_Id INT,
    H_Id INT,
    PRIMARY KEY (L_Id, H_Id),
    FOREIGN KEY (L_Id) REFERENCES Laundry_Service(L_Id),
    FOREIGN KEY (H_Id) REFERENCES Home_Service(H_Id)
);
```

Below the code, the message "Table created." is displayed, followed by "0.01 seconds".



For each created table inserted 5 rows of values:



```

User SCOTT
Home > SQL > SQL Commands
Autocommit Display: 10
Save Run
INSERT INTO Customer (C_Id, C_Name, C_Address, C_Email, C_Phone) VALUES (1, 'Tanvir Arabi', '123 Oak St, Anytown', 'john.s@email.com', '555-1234');
INSERT INTO Customer (C_Id, C_Name, C_Address, C_Email, C_Phone) VALUES (2, 'Bushra Kaiser', '456 Elm St, Anytown', 'jane.e@email.com', '555-5678');
INSERT INTO Customer (C_Id, C_Name, C_Address, C_Email, C_Phone) VALUES (3, 'Anas Khan', '789 Pine Ave, Anytown', 'p.jones@email.com', '555-9012');
INSERT INTO Customer (C_Id, C_Name, C_Address, C_Email, C_Phone) VALUES (4, 'Hridoy', '101 Maple Ln, Anytown', 'mary.b@email.com', '555-3456');
INSERT INTO Customer (C_Id, C_Name, C_Address, C_Email, C_Phone) VALUES (5, 'Arabi', '202 Birch Rd, Anytown', 'david.w@email.com', '555-7890');

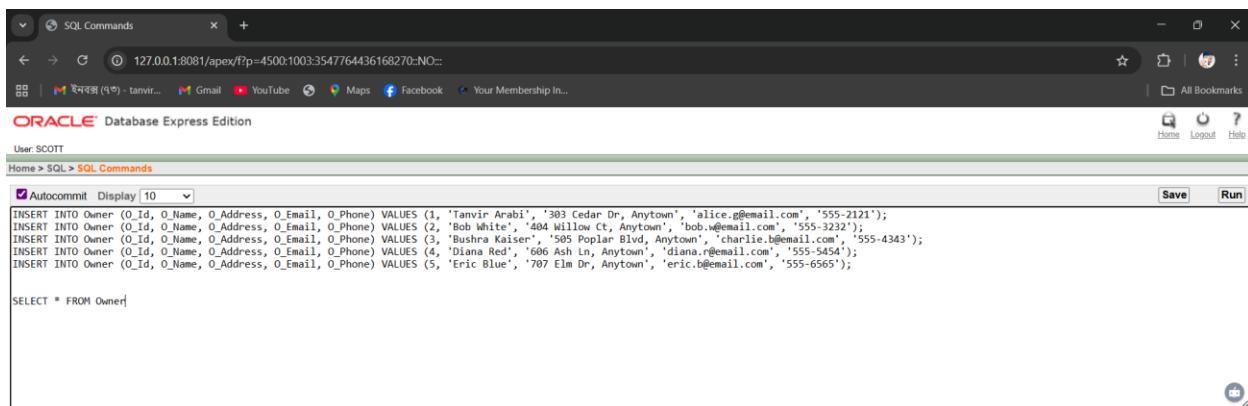
SELECT * FROM Customer

```

Results Explain Describe Saved SQL History

C_ID	C_NAME	C_ADDRESS	C_EMAIL	C_PHONE
1	Tanvir Arabi	123 Oak St, Anytown	john.s@email.com	555-1234
2	Bushra Kaiser	456 Elm St, Anytown	jane.e@email.com	555-5678
3	Anas Khan	789 Pine Ave, Anytown	p.jones@email.com	555-9012
4	Hridoy	101 Maple Ln, Anytown	mary.b@email.com	555-3456
5	Arabi	202 Birch Rd, Anytown	david.w@email.com	555-7890

5 rows returned in 0.03 seconds CSV Export

```

User SCOTT
Home > SQL > SQL Commands
Autocommit Display: 10
Save Run
INSERT INTO Owner (O_Id, O_Name, O_Address, O_Email, O_Phone) VALUES (1, 'Tanvir Arabi', '303 Cedar Dr, Anytown', 'alice.g@email.com', '555-2121');
INSERT INTO Owner (O_Id, O_Name, O_Address, O_Email, O_Phone) VALUES (2, 'Bob White', '404 Willow Ct, Anytown', 'bob.w@email.com', '555-3232');
INSERT INTO Owner (O_Id, O_Name, O_Address, O_Email, O_Phone) VALUES (3, 'Bushra Kaiser', '505 Poplar Blvd, Anytown', 'charlie.b@email.com', '555-4343');
INSERT INTO Owner (O_Id, O_Name, O_Address, O_Email, O_Phone) VALUES (4, 'Diana Red', '606 Ash Ln, Anytown', 'diana.r@email.com', '555-5454');
INSERT INTO Owner (O_Id, O_Name, O_Address, O_Email, O_Phone) VALUES (5, 'Eric Blue', '707 Elm Dr, Anytown', 'eric.b@email.com', '555-6565');

SELECT * FROM Owner

```

Results Explain Describe Saved SQL History

O_ID	O_NAME	O_ADDRESS	O_EMAIL	O_PHONE
1	Tanvir Arabi	303 Cedar Dr, Anytown	alice.g@email.com	555-2121
2	Bob White	404 Willow Ct, Anytown	bob.w@email.com	555-3232
3	Bushra Kaiser	505 Poplar Blvd, Anytown	charlie.b@email.com	555-4343
4	Diana Red	606 Ash Ln, Anytown	diana.r@email.com	555-5454
5	Eric Blue	707 Elm Dr, Anytown	eric.b@email.com	555-6565

5 rows returned in 0.00 seconds CSV Export



```

INSERT INTO Employees (E_Id, E_Name, E_DOB, E_Address, E_Phone) VALUES (1, 'Frank Miller', TO_DATE('1990-05-15', 'YYYY-MM-DD'), '808 Pine Ave, Anytown', '555-1122');
INSERT INTO Employees (E_Id, E_Name, E_DOB, E_Address, E_Phone) VALUES (2, 'Grace Lee', TO_DATE('1992-08-22', 'YYYY-MM-DD'), '909 Maple Rd, Anytown', '555-2345');
INSERT INTO Employees (E_Id, E_Name, E_DOB, E_Address, E_Phone) VALUES (3, 'Harry Kim', TO_DATE('1988-11-01', 'YYYY-MM-DD'), '111 Birch Ln, Anytown', '555-5566');
INSERT INTO Employees (E_Id, E_Name, E_DOB, E_Address, E_Phone) VALUES (4, 'Ivy Chen', TO_DATE('1995-02-10', 'YYYY-MM-DD'), '222 Oak Ln, Anytown', '555-7788');
INSERT INTO Employees (E_Id, E_Name, E_DOB, E_Address, E_Phone) VALUES (5, 'Jack Davis', TO_DATE('1991-07-28', 'YYYY-MM-DD'), '333 Elm Ct, Anytown', '555-9900');

SELECT * FROM Employees

```

E_ID	E_NAME	E_DOB	E_ADDRESS	E_PHONE
1	Frank Miller	15-MAY-90	808 Pine Ave, Anytown	555-1122
2	Grace Lee	22-AUG-92	909 Maple Rd, Anytown	555-3344
3	Harry Kim	01-NOV-88	111 Birch Ln, Anytown	555-5566
4	Ivy Chen	10-FEB-95	222 Oak Ln, Anytown	555-7788
5	Jack Davis	28-JUL-91	333 Elm Ct, Anytown	555-9900

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

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



```

INSERT INTO Laundry_Service (L_Id, L_Address, L_Phone, O_Id, E_Id) VALUES (1, '500 Main St, Anytown', '555-1001', 1, 1);
INSERT INTO Laundry_Service (L_Id, L_Address, L_Phone, O_Id, E_Id) VALUES (2, '600 Market St, Anytown', '555-2002', 2, 2);
INSERT INTO Laundry_Service (L_Id, L_Address, L_Phone, O_Id, E_Id) VALUES (3, '700 Broadway, Anytown', '555-3003', 3, 3);
INSERT INTO Laundry_Service (L_Id, L_Address, L_Phone, O_Id, E_Id) VALUES (4, '800 Park Ave, Anytown', '555-4004', 4, 4);
INSERT INTO Laundry_Service (L_Id, L_Address, L_Phone, O_Id, E_Id) VALUES (5, '900 Wall St, Anytown', '555-5005', 5, 5);

SELECT * FROM Laundry_Service

```

L_ID	L_ADDRESS	L_PHONE	O_ID	E_ID
1	500 Main St, Anytown	555-1001	1	1
2	600 Market St, Anytown	555-2002	2	2
3	700 Broadway, Anytown	555-3003	3	3
4	800 Park Ave, Anytown	555-4004	4	4
5	900 Wall St, Anytown	555-5005	5	5

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

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



SQL Commands

127.0.0.1:8081/apex/f?p=4500:1003:3547764436168270::NO::

User SCOTT

ORACLE Database Express Edition

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
INSERT INTO Home_Service (H_Id, H_Address, H_Pdate, H_Ddate) VALUES (1, '123 Oak St, Anytown', TO_DATE('2025-08-01', 'YYYY-MM-DD'), TO_DATE('2025-08-02', 'YYYY-MM-DD'));
INSERT INTO Home_Service (H_Id, H_Address, H_Pdate, H_Ddate) VALUES (2, '456 Elm St, Anytown', TO_DATE('2025-08-03', 'YYYY-MM-DD'), TO_DATE('2025-08-04', 'YYYY-MM-DD'));
INSERT INTO Home_Service (H_Id, H_Address, H_Pdate, H_Ddate) VALUES (3, '789 Pine Ave, Anytown', TO_DATE('2025-08-05', 'YYYY-MM-DD'), TO_DATE('2025-08-06', 'YYYY-MM-DD'));
INSERT INTO Home_Service (H_Id, H_Address, H_Pdate, H_Ddate) VALUES (4, '101 Maple Ln, Anytown', TO_DATE('2025-08-07', 'YYYY-MM-DD'), TO_DATE('2025-08-08', 'YYYY-MM-DD'));
INSERT INTO Home_Service (H_Id, H_Address, H_Pdate, H_Ddate) VALUES (5, '202 Birch Rd, Anytown', TO_DATE('2025-08-09', 'YYYY-MM-DD'), TO_DATE('2025-08-10', 'YYYY-MM-DD'));
```

SELECT * FROM Home_Service

Results Explain Describe Saved SQL History

H_ID	H_ADDRESS	H_PDATE	H_DDATE
1	123 Oak St, Anytown	01-AUG-25	02-AUG-25
2	456 Elm St, Anytown	03-AUG-25	04-AUG-25
3	789 Pine Ave, Anytown	05-AUG-25	06-AUG-25
4	101 Maple Ln, Anytown	07-AUG-25	08-AUG-25
5	202 Birch Rd, Anytown	09-AUG-25	10-AUG-25

5 rows returned in 0.01 seconds CSV Export



SQL Commands

127.0.0.1:8081/apex/f?p=4500:1003:3547764436168270::NO::

User SCOTT

ORACLE Database Express Edition

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
INSERT INTO Membership_Card (M_Id, M_Name, M_Phone, C_Id) VALUES (101, 'Gold', '555-1234', 1);
INSERT INTO Membership_Card (M_Id, M_Name, M_Phone, C_Id) VALUES (102, 'Silver', '555-5678', 2);
INSERT INTO Membership_Card (M_Id, M_Name, M_Phone, C_Id) VALUES (103, 'Bronze', '555-9012', 3);
INSERT INTO Membership_Card (M_Id, M_Name, M_Phone, C_Id) VALUES (104, 'Silver', '555-3456', 4);
INSERT INTO Membership_Card (M_Id, M_Name, M_Phone, C_Id) VALUES (105, 'Gold', '555-7890', 5);
```

SELECT * FROM Membership_Card

Results Explain Describe Saved SQL History

M_ID	M_NAME	M_PHONE	C_ID
101	Gold	555-1234	1
102	Silver	555-5678	2
103	Bronze	555-9012	3
104	Silver	555-3456	4
105	Gold	555-7890	5

5 rows returned in 0.00 seconds CSV Export



SQL Commands

127.0.0.1:8081/apex/f?p=4500:1003:3547764436168270::NO::

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit

```
INSERT INTO Utilities_Bill (U_Id, U_Paytype, U_due, L_Id) VALUES (1001, 'Credit Card', 25.50, 1);
INSERT INTO Utilities_Bill (U_Id, U_Paytype, U_due, L_Id) VALUES (1002, 'Debit Card', 30.75, 2);
INSERT INTO Utilities_Bill (U_Id, U_Paytype, U_due, L_Id) VALUES (1003, 'Cash', 15.00, 3);
INSERT INTO Utilities_Bill (U_Id, U_Paytype, U_due, L_Id) VALUES (1004, 'Credit Card', 45.20, 4);
INSERT INTO Utilities_Bill (U_Id, U_Paytype, U_due, L_Id) VALUES (1005, 'Debit Card', 50.00, 5);
```

SELECT * FROM Utilities_Bill

Results Explain Describe Saved SQL History

U_ID	U_PAYTYPE	U_DUE	L_ID
1001	Credit Card	25.5	1
1002	Debit Card	30.75	2
1003	Cash	15	3
1004	Credit Card	45.2	4
1005	Debit Card	50	5

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

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

SQL Commands

127.0.0.1:8081/apex/f?p=4500:1003:3547764436168270::NO::

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit

```
INSERT INTO Cust_Pays_Bill (C_Id, U_Id) VALUES (1, 1001);
INSERT INTO Cust_Pays_Bill (C_Id, U_Id) VALUES (2, 1002);
INSERT INTO Cust_Pays_Bill (C_Id, U_Id) VALUES (3, 1003);
INSERT INTO Cust_Pays_Bill (C_Id, U_Id) VALUES (4, 1004);
INSERT INTO Cust_Pays_Bill (C_Id, U_Id) VALUES (5, 1005);
```

SELECT * FROM Cust_Pays_Bill

Results Explain Describe Saved SQL History

C_ID	U_ID
1	1001
2	1002
3	1003
4	1004
5	1005

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

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

User SCOTT

Home > SQL > SQL Commands

```
Autocommit Display 10
INSERT INTO LS_Offers_HS (L_Id, H_Id) VALUES (1, 1);
INSERT INTO LS_Offers_HS (L_Id, H_Id) VALUES (2, 2);
INSERT INTO LS_Offers_HS (L_Id, H_Id) VALUES (3, 3);
INSERT INTO LS_Offers_HS (L_Id, H_Id) VALUES (4, 4);
INSERT INTO LS_Offers_HS (L_Id, H_Id) VALUES (5, 5);

SELECT * FROM LS_Offers_HS
```

Results Explain Describe Saved SQL History

L_ID	H_ID
1	1
2	2
3	3
4	4
5	5

5 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.

Basic PL/SQL

1. Variables and Operators

User SCOTT

Home > SQL > SQL Commands

```
Autocommit Display 10
DECLARE
    v_laundry_service_id NUMBER := 1; -- Variable 1
    v_home_service_id NUMBER := 2; -- Variable 2
    v_total_value NUMBER;
    v_product_value NUMBER;
BEGIN
    -- Operator 1 (+)
    v_total_value := v_laundry_service_id + v_home_service_id;
    DBMS_OUTPUT.PUT_LINE('The sum of Laundry Service ID and Home Service ID is: ' || v_total_value);

    -- Operator 2 (*)
    v_product_value := v_laundry_service_id * v_home_service_id;
    DBMS_OUTPUT.PUT_LINE('The product of Laundry Service ID and Home Service ID is: ' || v_product_value);
END;
/
```

The sum of Laundry Service ID and Home Service ID is: 3
The product of Laundry Service ID and Home Service ID is: 2

Statement processed.

0.09 seconds

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

2. Single-Row Functions

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

```
DECLARE
    v_home_address VARCHAR2(255);
    v_home_service_id NUMBER := 3;
    v_laundry_service_address VARCHAR2(255);
    v_laundry_service_id NUMBER := 2;
BEGIN
    -- Function 1: UPPER
    SELECT UPPER(H.Address) INTO v_home_address FROM Home_Service WHERE H_Id = v_home_service_id;
    DBMS_OUTPUT.PUT_LINE('Uppercase address: ' || v_home_address);

    -- Function 2: SUBSTR
    SELECT SUBSTR(L_Address, 1, 10) INTO v_laundry_service_address FROM Laundry_Service WHERE L_Id = v_laundry_service_id;
    DBMS_OUTPUT.PUT_LINE('Substring of laundry address: ' || v_laundry_service_address);
END;
/
```

The results section shows the output of the code:

```
Uppercase address: 789 PINE AVE, ANYTOWN
Substring of laundry address: 600 Market
```

Statement processed.

0.04 seconds

Language: en-gb Application Express 2.1 0.00 39 Copyright © 1999, 2006, Oracle. All rights reserved.

3. Group Functions

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

```
DECLARE
    v_average_due DECIMAL(10, 2);
    v_total_bills NUMBER;
    v_total_due DECIMAL(10, 2);
BEGIN
    -- Function 1: AVG
    SELECT AVG(U.due) INTO v_average_due FROM Utilities_Bill;
    DBMS_OUTPUT.PUT_LINE('Average bill amount: $' || v_average_due);

    -- Function 2: SUM
    SELECT SUM(U.due) INTO v_total_due FROM Utilities_Bill;
    DBMS_OUTPUT.PUT_LINE('Total bill amount: $' || v_total_due);
END;
/
```

The results section shows the output of the code:

```
Average bill amount: $33.29
Total bill amount: $166.45
```

Statement processed.

0.02 seconds

Language: en-gb Application Express 2.1 0.00 39 Copyright © 1999, 2006, Oracle. All rights reserved.

4. Loop



```
DECLARE
    v_counter NUMBER := 1;
BEGIN
    LOOP
        DBMS_OUTPUT.PUT_LINE('Processing Laundry Service with ID: ' || v_counter);
        v_counter := v_counter + 1;
        EXIT WHEN v_counter > 5;
    END LOOP;
END;
```

-- Example 2: FOR loop to iterate through all customers.

```
BEGIN
    FOR rec IN (SELECT C_Id, C_Name FROM Customer) LOOP
        DBMS_OUTPUT.PUT_LINE('Customer ' || rec.C_Id || ': ' || rec.C_Name);
    END LOOP;
END;
```

Results Explain Describe Saved SQL History

Processing Laundry Service with ID: 1
Processing Laundry Service with ID: 2
Processing Laundry Service with ID: 3
Processing Laundry Service with ID: 4
Processing Laundry Service with ID: 5
Statement processed.

0.00 seconds

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

```
DECLARE
    v_counter NUMBER := 1;
BEGIN
    LOOP
        DBMS_OUTPUT.PUT_LINE('Processing Laundry Service with ID: ' || v_counter);
        v_counter := v_counter + 1;
        EXIT WHEN v_counter > 5;
    END LOOP;
END;
```

-- Example 2: FOR loop to iterate through all customers.

```
BEGIN
    FOR rec IN (SELECT C_Id, C_Name FROM Customer) LOOP
        DBMS_OUTPUT.PUT_LINE('Customer ' || rec.C_Id || ': ' || rec.C_Name);
    END LOOP;
END;
```

Results Explain Describe Saved SQL History

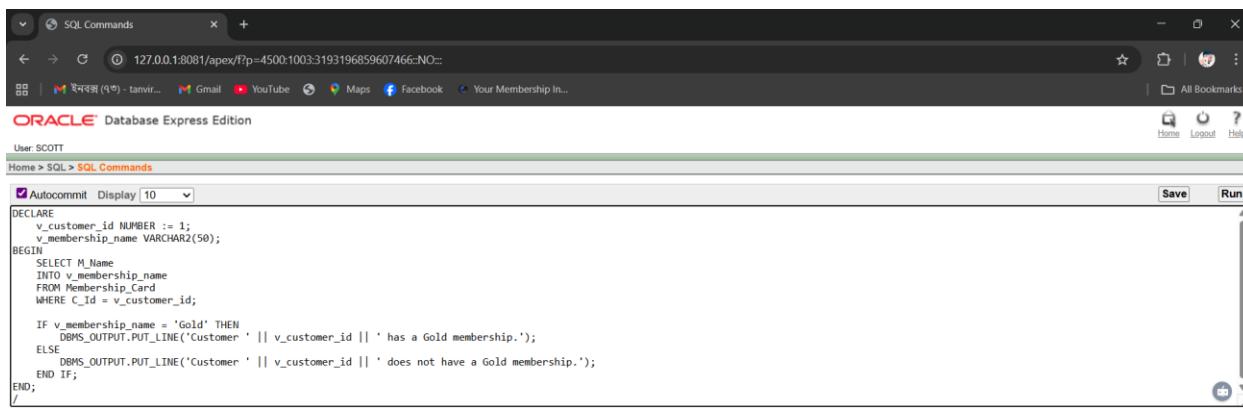
Customer 1: Tanvir Arabi
Customer 2: Bushra Kaiser
Customer 3: Anas Khan
Customer 4: Hridoy
Customer 5: Arabi
Customer 6: Tony Stark
Statement processed.

0.02 seconds

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



5. Conditional Statements



```
DECLARE
    v_customer_id NUMBER := 1;
    v_membership_name VARCHAR2(50);
BEGIN
    SELECT M_Name
    INTO v_membership_name
    FROM Membership_Card
    WHERE C_Id = v_customer_id;

    IF v_membership_name = 'Gold' THEN
        DBMS_OUTPUT.PUT_LINE('Customer ' || v_customer_id || ' has a Gold membership.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Customer ' || v_customer_id || ' does not have a Gold membership.');
    END IF;
END;
/
```

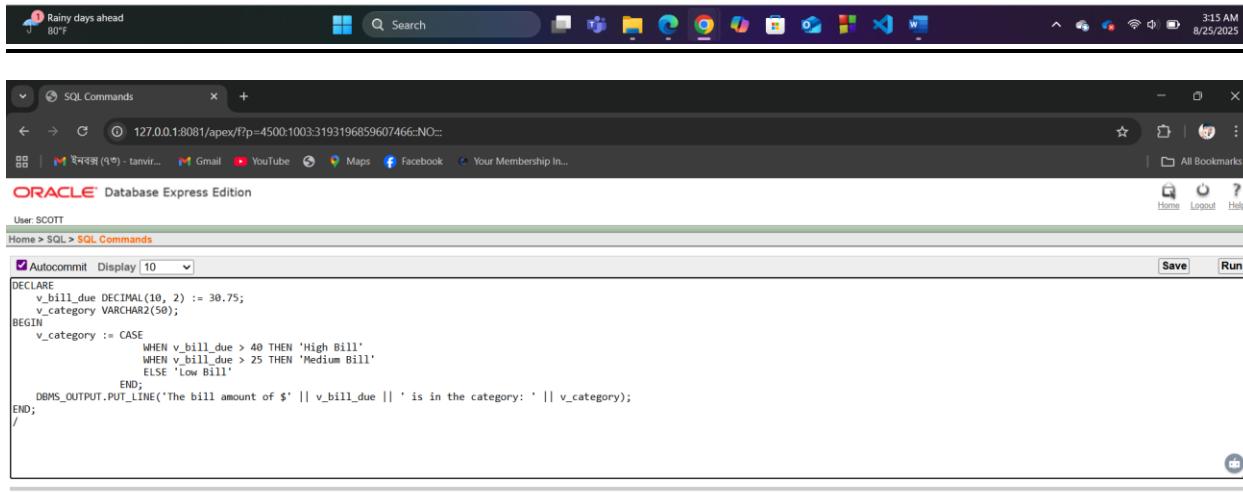
Results Explain Describe Saved SQL History

Customer 1 has a Gold membership.

Statement processed.

0.03 seconds

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



```
DECLARE
    v_bill_due DECIMAL(10, 2) := 30.75;
    v_category VARCHAR2(50);
BEGIN
    v_category := CASE
        WHEN v_bill_due > 40 THEN 'High Bill'
        WHEN v_bill_due > 25 THEN 'Medium Bill'
        ELSE 'Low Bill'
    END;
    DBMS_OUTPUT.PUT_LINE('The bill amount of $' || v_bill_due || ' is in the category: ' || v_category);
END;
/
```

Results Explain Describe Saved SQL History

The bill amount of \$30.75 is in the category: Medium Bill

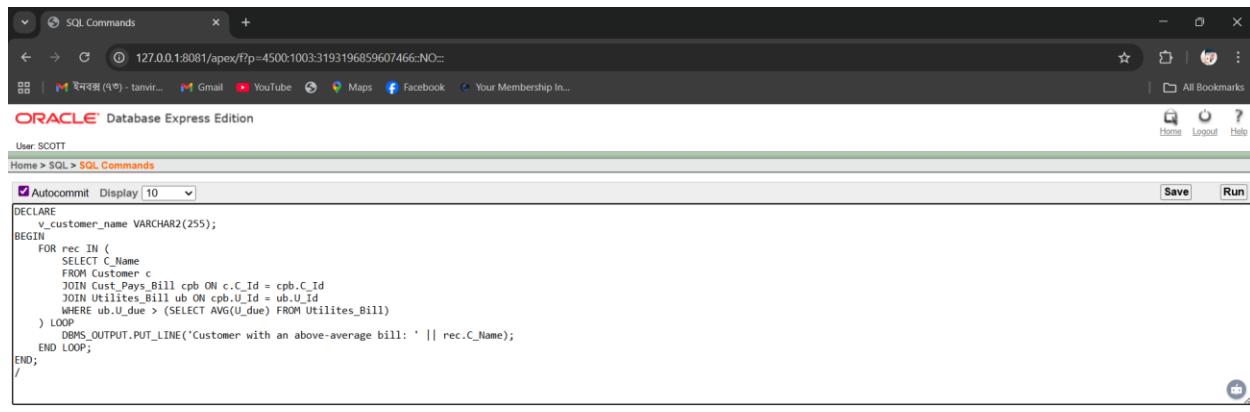
Statement processed.

0.01 seconds

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



6. Subquery



```
DECLARE
  v_customer_name VARCHAR2(255);
BEGIN
  FOR rec IN (
    SELECT C.Name
    FROM Customer C
    JOIN Cust_Pays_Bill cpb ON c.C_Id = cpb.C_Id
    JOIN Utilities_Bill ub ON cpb.U_Id = ub.U_Id
    WHERE ub.U_due > (SELECT AVG(U_due) FROM Utilities_Bill)
  ) LOOP
    DBMS_OUTPUT.PUT_LINE('Customer with an above-average bill: ' || rec.C_Name);
  END LOOP;
END;
/
```

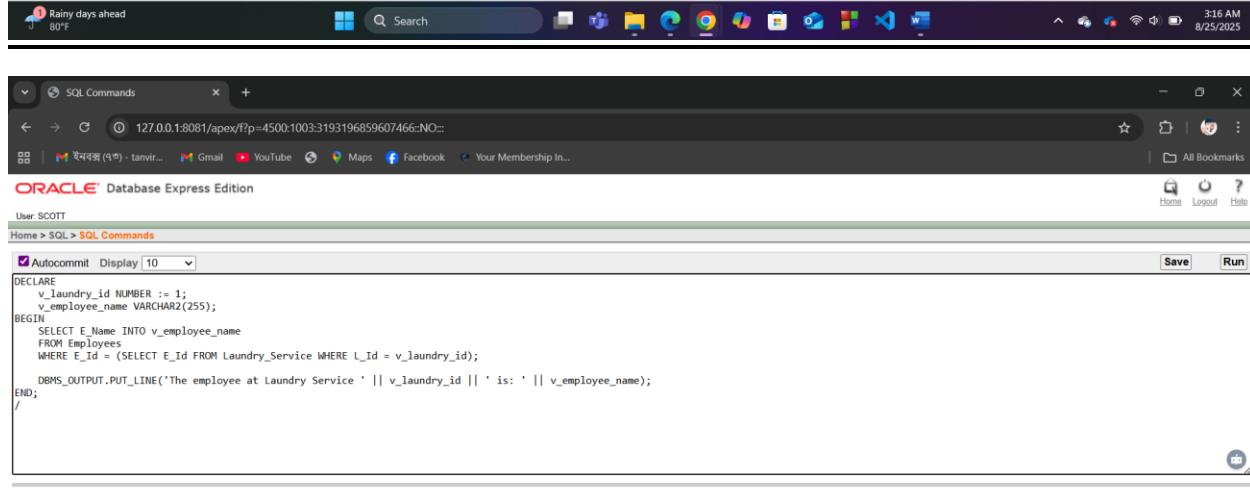
Results Explain Describe Saved SQL History

Customer with an above-average bill: Hridoy
Customer with an above-average bill: Arabi

Statement processed.

0.05 seconds

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



```
DECLARE
  v_laundry_id NUMBER := 1;
  v_employee_name VARCHAR2(255);
BEGIN
  SELECT E_Name INTO v_employee_name
  FROM Employees
  WHERE E_Id = (SELECT E_Id FROM Laundry_Service WHERE L_Id = v_laundry_id);

  DBMS_OUTPUT.PUT_LINE('The employee at Laundry Service ' || v_laundry_id || ' is: ' || v_employee_name);
END;
/
```

Results Explain Describe Saved SQL History

The employee at Laundry Service 1 is: Frank Miller Jr.

Statement processed.

0.02 seconds

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



7. Joining

The screenshot shows the Oracle Database Express Edition interface. A PL/SQL script is being run to print customer bills. The output shows five rows of bill information.

```
BEGIN
FOR rec IN (
    SELECT c.C_Name, ub.U_Id, ub.U_due
    FROM Customer c
    JOIN Cust_Pays_Bill cpb ON c.C_Id = cpb.C_Id
    JOIN Utilities_Bill ub ON cpb.U_Id = ub.U_Id
) LOOP
DBMS_OUTPUT.PUT_LINE(rec.C_Name || ' paid bill ' || rec.U_Id || ' for $' || rec.U_due);
END LOOP;
END;
```

Tanvir Arabi paid bill 1001 for \$25.5
Bushra Kaiser paid bill 1002 for \$30.75
Anas Khan paid bill 1003 for \$15
Hridoy paid bill 1004 for \$45.2
Arabi paid bill 1005 for \$50

Statement processed.

0.02 seconds

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

The screenshot shows the Oracle Database Express Edition interface. A PL/SQL script is being run to print laundry service details. The output shows five rows of service information.

```
BEGIN
FOR rec IN (
    SELECT ls.L_Address, o.O_Name
    FROM Laundry_Service ls
    JOIN Owner o ON ls.O_Id = o.O_Id
) LOOP
DBMS_OUTPUT.PUT_LINE('The laundry service at ' || rec.L_Address || ' is owned by ' || rec.O_Name);
END LOOP;
END;
```

The laundry service at 900 Main St, Anytown is owned by Tanvir Arabi
The laundry service at 800 Market St, Anytown is owned by Bob White
The laundry service at 700 Broadway, Anytown is owned by Bushra Kaiser
The laundry service at 800 Park Ave, Anytown is owned by Diana Red
The laundry service at 900 Wall St, Anytown is owned by Eric Blue
The laundry service at 888 Main St is owned by Tanvir Arabi

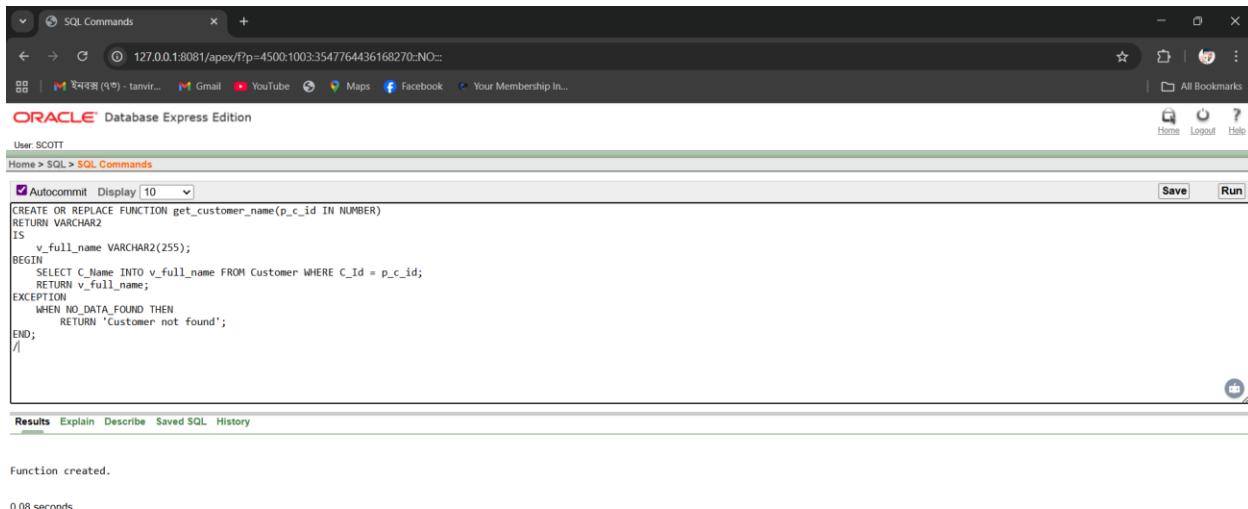
Statement processed.

0.03 seconds

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

ADVANCE PL/SQL

1. Stored Function



SQL Commands

127.0.0.1:8081/apex/f?p=4500:1003:3547764436168270::NO::

User SCOTT

ORACLE Database Express Edition

Home > SQL > SQL Commands

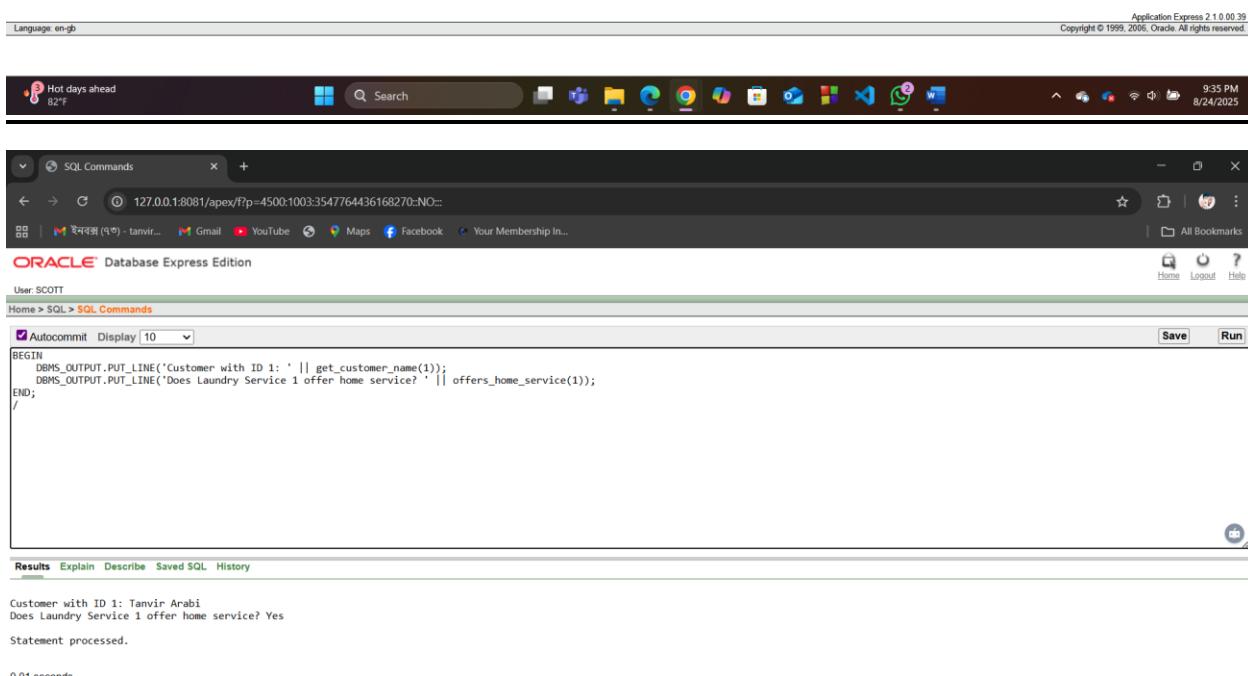
Autocommit Display 10 Save Run

```
CREATE OR REPLACE FUNCTION get_customer_name(p_c_id IN NUMBER)
RETURN VARCHAR2
IS
    v_full_name VARCHAR2(255);
BEGIN
    SELECT C_Name INTO v_full_name FROM Customer WHERE C_Id = p_c_id;
    RETURN v_full_name;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN 'Customer not found';
END;
/
```

Results Explain Describe Saved SQL History

Function created.

0.08 seconds



Application Express 2.1 0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-gb

Hot days ahead 82°F

Search

9:35 PM 8/24/2025

SQL Commands

127.0.0.1:8081/apex/f?p=4500:1003:3547764436168270::NO::

User SCOTT

ORACLE Database Express Edition

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
BEGIN
    DBMS_OUTPUT.PUT_LINE('Customer with ID 1: ' || get_customer_name(1));
    DBMS_OUTPUT.PUT_LINE('Does Laundry Service 1 offer home service? ' || offers_home_service(1));
END;
/
```

Results Explain Describe Saved SQL History

Customer with ID 1: Tanvir Arabi
Does Laundry Service 1 offer home service? Yes
Statement processed.

0.01 seconds



Application Express 2.1 0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-gb

Rainy days ahead 82°F

Search

9:36 PM 8/24/2025

2. Stored Procedure

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, a stored procedure named `add_home_service` is being created. The code defines a procedure that takes four parameters: `p_h_id`, `p_h_address`, `p_h_pdate`, and `p_h_ddate`. It inserts a new row into the `Home_Service` table with these values and outputs a message to the DBMS_OUTPUT. The procedure ends with a closing brace. Below the code, the message "Procedure created." is displayed, followed by "0.02 seconds".

```
CREATE OR REPLACE PROCEDURE add_home_service(
    p_h_id IN NUMBER,
    p_h_address IN VARCHAR2,
    p_h_pdate IN DATE,
    p_h_ddate IN DATE
)
IS
BEGIN
    INSERT INTO Home_Service (H_Id, H_Address, H_Pdate, H_Ddate)
    VALUES (p_h_id, p_h_address, p_h_pdate, p_h_ddate);
    DBMS_OUTPUT.PUT_LINE('New home service added with ID: ' || p_h_id);
END;
/
```



The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, a stored procedure named `update_customer_phone` is being created. The code defines a procedure that updates the `C_Phone` field in the `Customer` table for a specific customer identified by `p_c_id`. It also outputs a message to the DBMS_OUTPUT. The procedure ends with a closing brace. Below the code, the message "Procedure created." is displayed, followed by "0.02 seconds".

```
CREATE OR REPLACE PROCEDURE update_customer_phone(
    p_c_id IN NUMBER,
    p_new_phone IN VARCHAR2
)
IS
BEGIN
    UPDATE Customer SET C_Phone = p_new_phone WHERE C_Id = p_c_id;
    DBMS_OUTPUT.PUT_LINE('Phone number for customer ' || p_c_id || ' updated.');
END;
/
```



User SCOTT

Home > SQL > SQL Commands

```
BEGIN
    update_customer_phone(1, '555-9876');
    add_home_service(6, '999 New St, Anytown', TO_DATE('2025-08-11', 'YYYY-MM-DD'), TO_DATE('2025-08-12', 'YYYY-MM-DD'));
END;
/
```

Save Run

Results Explain Describe Saved SQL History

Phone number for customer 1 updated.
New home service added with ID: 6
Statement processed.

0.01 seconds

Application Express 21.0.0.39
Copyright © 1995, 2006, Oracle. All rights reserved.

Language: en-gb

82°F Haze

Search

9:39 PM 8/24/2025

3. Table-Based Record

User SCOTT

Home > SQL > SQL Commands

```
DECLARE
    v_customer_rec Customer%ROWTYPE;
BEGIN
    SELECT * INTO v_customer_rec FROM Customer WHERE C_Id = 1;
    DBMS_OUTPUT.PUT_LINE('Customer Record: ' || v_customer_rec.C_Name || ', ' || v_customer_rec.C_Email);
END;
/
```

Save Run

Results Explain Describe Saved SQL History

Customer Record: Tanvir Arabi, john.s@email.com
Statement processed.

0.02 seconds

Application Express 21.0.0.39
Copyright © 1995, 2006, Oracle. All rights reserved.

Language: en-gb

82°F Haze

Search

9:40 PM 8/24/2025

```

DECLARE
  v_employee_rec Employees%ROWTYPE;
BEGIN
  SELECT * INTO v_employee_rec FROM Employees WHERE E_Id = 1;
  DBMS_OUTPUT.PUT_LINE('Employee Record: ' || v_employee_rec.E_Name || ', ' || v_employee_rec.E_Phone);
END;
/

```

Employee Record: Frank Miller, 555-1122
Statement processed.

0.00 seconds

Language: en-gb Application Express 21.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

4. Explicit Curso

```

CURSOR c_expensive_bills IS
  SELECT U.Id, U.due
  FROM Utilities.Bill
  WHERE U.due > 30;
  v_bill_id NUMBER;
  v_bill_due DECIMAL(10, 2);
BEGIN
  OPEN c_expensive_bills;
  LOOP
    FETCH c_expensive_bills INTO v_bill_id, v_bill_due;
    EXIT WHEN c_expensive_bills%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Expensive Bill ID: ' || v_bill_id || ', Amount: $' || v_bill_due);
  END LOOP;
  CLOSE c_expensive_bills;
END;
/

```

Expensive Bill ID: 1002, Amount: \$30.75
Expensive Bill ID: 1004, Amount: \$45.2
Expensive Bill ID: 1005, Amount: \$50

Statement processed.

0.03 seconds

Language: en-gb Application Express 21.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

ORAQUEL Database Express Edition

User SCOTT

Home > SQL > SQL Commands

```

 Autocommit Display 10
SELECT ls.L_Id, ls.L_Address, o.O_Name
FROM Laundry_Service ls
JOIN Owner o ON ls.O_Id = o.O_Id;
v_laundry_id NUMBER;
v_laundry_address VARCHAR2(255);
v_owner_name VARCHAR2(255);
BEGIN
OPEN c_laundry_services;
LOOP
  FETCH c_laundry_services INTO v_laundry_id, v_laundry_address, v_owner_name;
  EXIT WHEN c_laundry_services%NOTFOUND;
  DBMS_OUTPUT.PUT_LINE('Laundry Service ' || v_laundry_id || ' at ' || v_laundry_address || ' owned by ' || v_owner_name);
END LOOP;
CLOSE c_laundry_services;
END;
/

```

Results **Explain** **Describe** **Saved SQL** **History**

Laundry Service 1 at 500 Main St, Anytown owned by Tanvir Arabi
Laundry Service 2 at 600 Market St, Anytown owned by Bob White
Laundry Service 3 at 700 Broadway, Anytown owned by Bushra Kaiser
Laundry Service 4 at 800 Park Ave, Anytown owned by Diana Red
Laundry Service 5 at 900 Wall St, Anytown owned by Eric Blue

Statement processed.

0.02 seconds

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

5. Cursor-Based Record

ORAQUEL Database Express Edition

User SCOTT

Home > SQL > SQL Commands

```

 Autocommit Display 10
DECLARE
CURSOR c_customers IS SELECT C_Id, C_Name, C_Address FROM Customer;
v_customer_rec c_customers%ROWTYPE;
BEGIN
OPEN c_customers;
LOOP
  FETCH c_customers INTO v_customer_rec;
  EXIT WHEN c_customers%NOTFOUND;
  DBMS_OUTPUT.PUT_LINE('Customer ID: ' || v_customer_rec.C_Id || ', Name: ' || v_customer_rec.C_Name);
END LOOP;
CLOSE c_customers;
END;
/

```

Results **Explain** **Describe** **Saved SQL** **History**

Customer ID: 1, Name: Tanvir Arabi
Customer ID: 2, Name: Bushra Kaiser
Customer ID: 3, Name: Anas Khan
Customer ID: 4, Name: Hridya
Customer ID: 5, Name: Arabi

Statement processed.

0.02 seconds

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

```

DECLARE
  CURSOR c_home_services IS SELECT H_Id, H_Address, H_Pdate FROM Home_Service;
  v_home_rec c_home_services%ROWTYPE;
BEGIN
  FOR v_home_rec IN c_home_services LOOP
    DBMS_OUTPUT.PUT_LINE('Home Service ID: ' || v_home_rec.H_Id || ', Address: ' || v_home_rec.H_Address);
  END LOOP;
END;
/

```

Results Explain Describe Saved SQL History

Home Service ID: 1, Address: 123 Oak St, Anytown
Home Service ID: 2, Address: 456 Elm St, Anytown
Home Service ID: 3, Address: 789 Pine Ave, Anytown
Home Service ID: 4, Address: 101 Maple Ln, Anytown
Home Service ID: 5, Address: 202 Birch Rd, Anytown
Home Service ID: 6, Address: 999 New St, Anytown

Statement processed.

0.02 seconds

Application Express 21.0.0.39
Copyright © 1995, 2006, Oracle. All rights reserved.

Language: en-gb

6.. Row-Level Trigger (2 examples)

```

CREATE OR REPLACE TRIGGER default_customer_phone
BEFORE INSERT ON Customer
FOR EACH ROW
BEGIN
  IF :NEW.C_Phone IS NULL THEN
    :NEW.C_Phone := '999-9999';
  END IF;
END;
/
-- Example usage to fire the triggers
BEGIN
  UPDATE Employees SET E_Name = 'Frank Miller Jr.' WHERE E_Id = 1;
  INSERT INTO Customer (C_Id, C_Name, C_Address, C_Email) VALUES (6, 'Tony Stark', '1 Iron Man Dr', 'tony.s@email.com');
END;
/

```

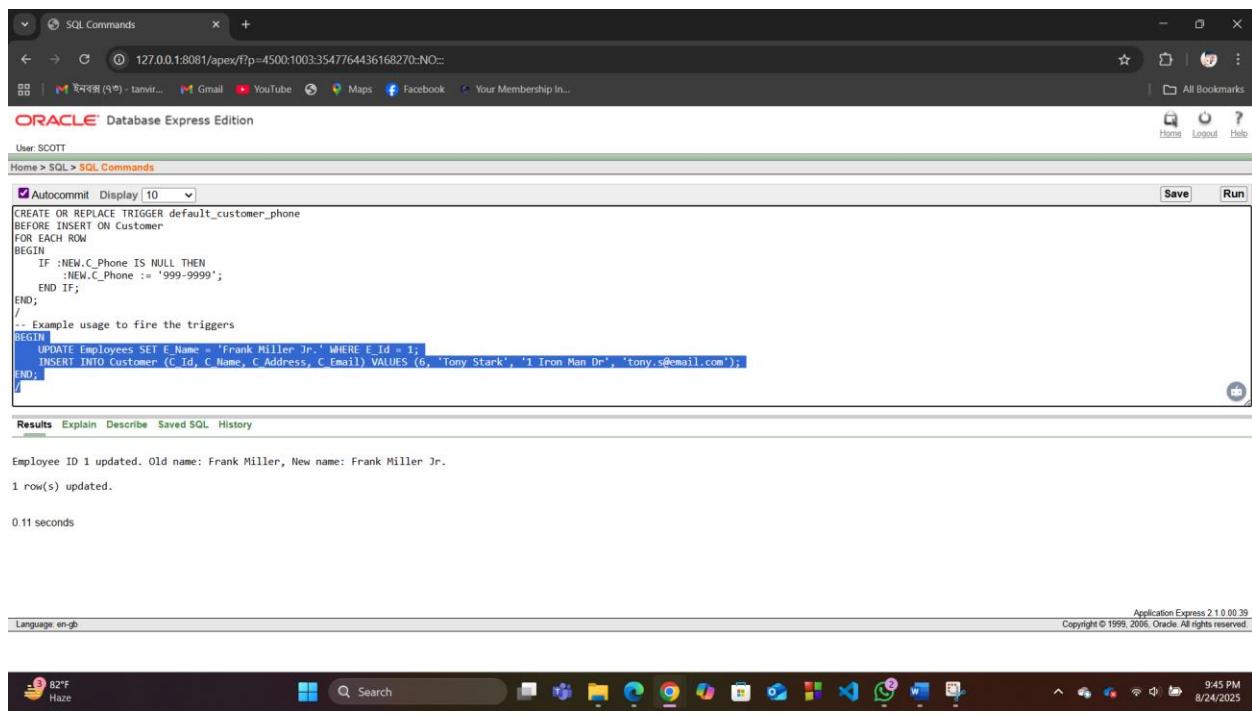
Results Explain Describe Saved SQL History

Trigger created.

0.01 seconds

Application Express 21.0.0.39
Copyright © 1995, 2006, Oracle. All rights reserved.

Language: en-gb



User SCOTT

Home > SQL > SQL Commands

```
CREATE OR REPLACE TRIGGER default_customer_phone
BEFORE INSERT ON Customer
FOR EACH ROW
BEGIN
    IF :NEW.C_Phone IS NULL THEN
        :NEW.C_Phone := '999-9999';
    END IF;
END;
/
-- Example usage to fire the triggers
BEGIN
    UPDATE Employees SET E_Name = 'Frank Miller Jr.' WHERE E_Id = 1;
    INSERT INTO Customer (C_Id, C_Name, C_Address, C_Email) VALUES (6, 'Tony Stark', '1 Iron Man Dr', 'tony.s@email.com');
END;
/
```

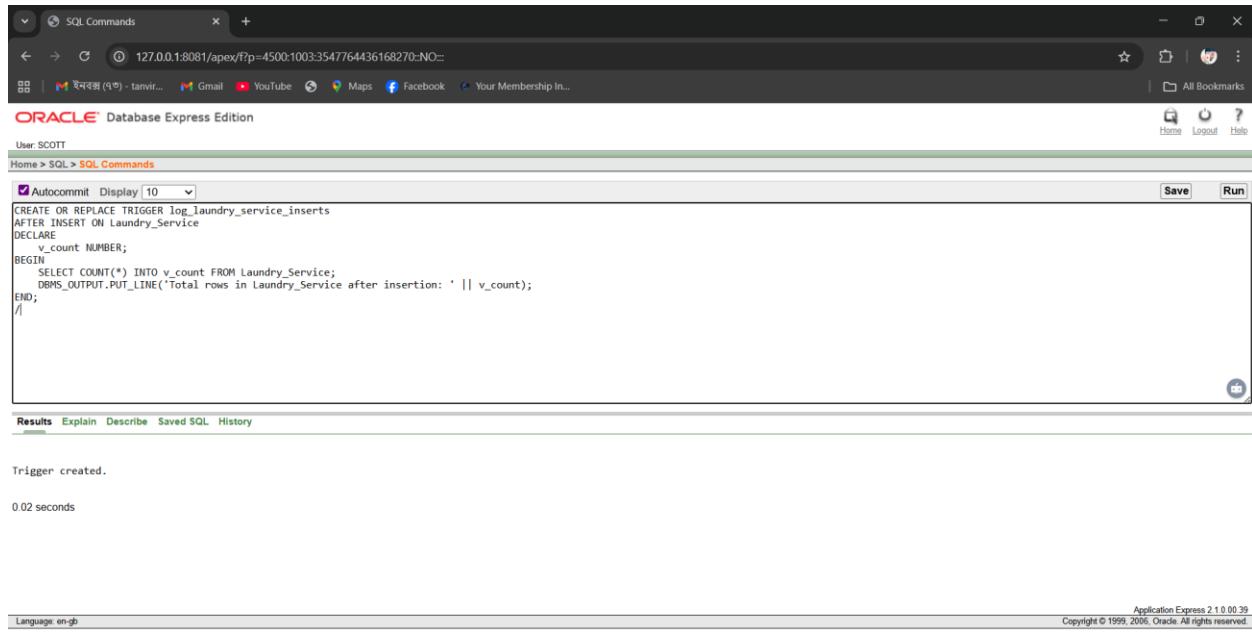
Results Explain Describe Saved SQL History

Employee ID 1 updated. Old name: Frank Miller, New name: Frank Miller Jr.
1 row(s) updated.

0.11 seconds

Application Express 21.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

7. Statement-Level Trigger



User SCOTT

Home > SQL > SQL Commands

```
CREATE OR REPLACE TRIGGER log_laundry_service_inserts
AFTER INSERT ON Laundry_Service
DECLARE
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count FROM Laundry_Service;
    DBMS_OUTPUT.PUT_LINE('Total rows in Laundry_Service after insertion: ' || v_count);
END;
/
```

Results Explain Describe Saved SQL History

Trigger created.
0.02 seconds

Application Express 21.0.0.39
Copyright © 1999, 2006, Oracle. All rights reserved.

User SCOTT

Home > SQL > SQL Commands

```
CREATE OR REPLACE TRIGGER log_home_service_deletes
AFTER DELETE ON Home_Service
BEGIN
  DBMS_OUTPUT.PUT_LINE('A DELETE statement was executed on the Home_Service table.');
END;
/
```

Save Run

Trigger created.

0.00 seconds



User SCOTT

Home > SQL > SQL Commands

```
BEGIN
  INSERT INTO Laundry_Service (L_Id, L_Address, L_Phone, O_Id, E_Id) VALUES (6, '888 Main St', '555-8888', 1, 1);
  DELETE FROM Home_Service WHERE H_Id = 6;
END;
/
```

Total rows in Laundry_Service after insertion: 6
A DELETE statement was executed on the Home_Service table.

1 row(s) inserted.

0.03 seconds



8. Package

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, a package named 'laundry_service_pkg' is being created. The code defines a procedure 'add_employee' that takes parameters for employee ID, name, date of birth, address, and phone number. It also includes a function 'get_employee_count' that returns the number of employees. The package is named 'laundry_service_pkg'. The results show the package was created successfully.

```
CREATE OR REPLACE PACKAGE laundry_service_pkg IS
    PROCEDURE add_employee(
        p_e_id IN NUMBER,
        p_e_name IN VARCHAR2,
        p_e_dob IN DATE,
        p_e_address IN VARCHAR2,
        p_e_phone IN VARCHAR2
    );
    -- Function to get the number of employees
    FUNCTION get_employee_count RETURN NUMBER;
END laundry_service_pkg;
/
```

Package created.
0.03 seconds

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



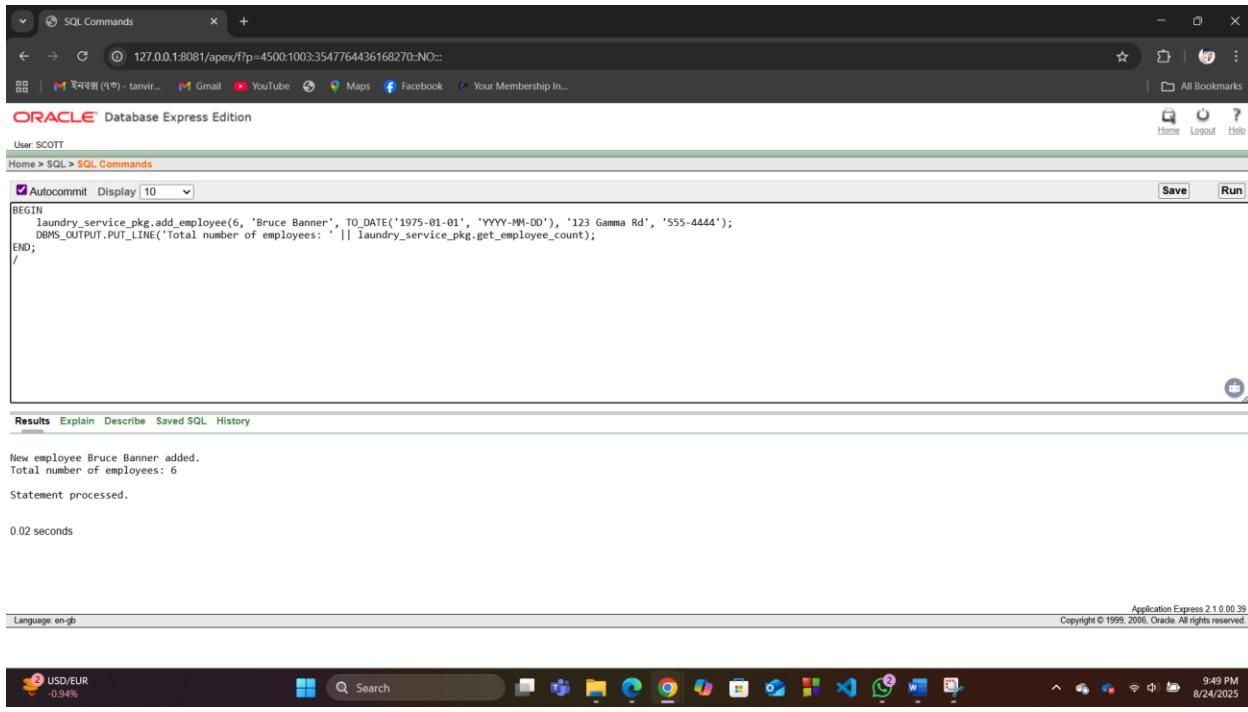
The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, a package body named 'laundry_service_pkg' is being created. It contains the same procedure 'add_employee' and function 'get_employee_count' as the previous package. Additionally, it includes an 'IS' block that inserts data into the 'Employees' table and outputs a message using 'DBMS_OUTPUT.PUT_LINE'. The package body is named 'laundry_service_pkg'. The results show the package body was created successfully.

```
CREATE OR REPLACE PACKAGE BODY laundry_service_pkg IS
    PROCEDURE add_employee(
        p_e_id IN NUMBER,
        p_e_name IN VARCHAR2,
        p_e_dob IN DATE,
        p_e_address IN VARCHAR2,
        p_e_phone IN VARCHAR2
    ) IS
    BEGIN
        INSERT INTO Employees (E_Id, E_Name, E_DOB, E_Address, E_Phone)
        VALUES (p_e_id, p_e_name, p_e_dob, p_e_address, p_e_phone);
        DBMS_OUTPUT.PUT_LINE('New employee ' || p_e_name || ' added.');
    END;
    FUNCTION get_employee_count RETURN NUMBER IS
        v_count NUMBER;
    BEGIN
        SELECT COUNT(*) INTO v_count FROM Employees;
        RETURN v_count;
    END;
END laundry_service_pkg;
/
```

Package Body created.
0.02 seconds

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





The screenshot shows the Oracle Database Express Edition SQL Commands interface. A SQL command is run to add an employee named Bruce Banner with ID 6, hired on January 1, 1975, at address 123 Gamma Rd, and phone number 555-4444. The command also outputs the total number of employees, which is 6. The results show the new employee added and the total count updated.

```

SQL Commands
127.0.0.1:8081/apex/f?p=4500:1003:3547764436168270::NO::
ORACLE Database Express Edition
User: SCOTT
Home > SQL > SQL Commands
Autocommit Display: 10
BEGIN
  laundry_service_pkg.add_employee(6, 'Bruce Banner', TO_DATE('1975-01-01', 'YYYY-MM-DD'), '123 Gamma Rd', '555-4444');
  DBMS_OUTPUT.PUT_LINE('Total number of employees: ' || laundry_service_pkg.get_employee_count());
END;
/

```

Results Explain Describe Saved SQL History

New employee Bruce Banner added.
Total number of employees: 6
Statement processed.
0.02 seconds

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

Conclusion:

The Laundry Service Management System is currently an effective tool for boosting operational efficiency and customer satisfaction through automation and features like home service tracking. The proposed future work outlines a strategic evolution, beginning with integrating payment gateways and advanced analytics for immediate impact, followed by developing a mobile application to enhance accessibility and service delivery. The long-term vision involves incorporating AI for predictive maintenance and demand forecasting, along with sustainability tracking, to transform the system into an innovative, data-driven platform that ensures scalable growth and a significant competitive advantage in the market.