HOSTEL MANAGEMENT SYSTEM

Name:	**Muhammad Akhtar Nadeem**
Roll No:	**058966**

Week 1-3: Proposal Submission and Initial Setup

Tasks:

1. **Submission of project proposal, ERD, and database schema.**
2. **Database creation and implementation of tables.**
3. **Basic CRUD operations performed on the tables.**

1. Submission of project proposal, ERD, and database schema.

Title: Hostel Management System

Scenario

The Hostel Management System is designed to streamline the management of university hostels by efficiently handling student allocation, room assignments, attendance tracking, and administrative roles.

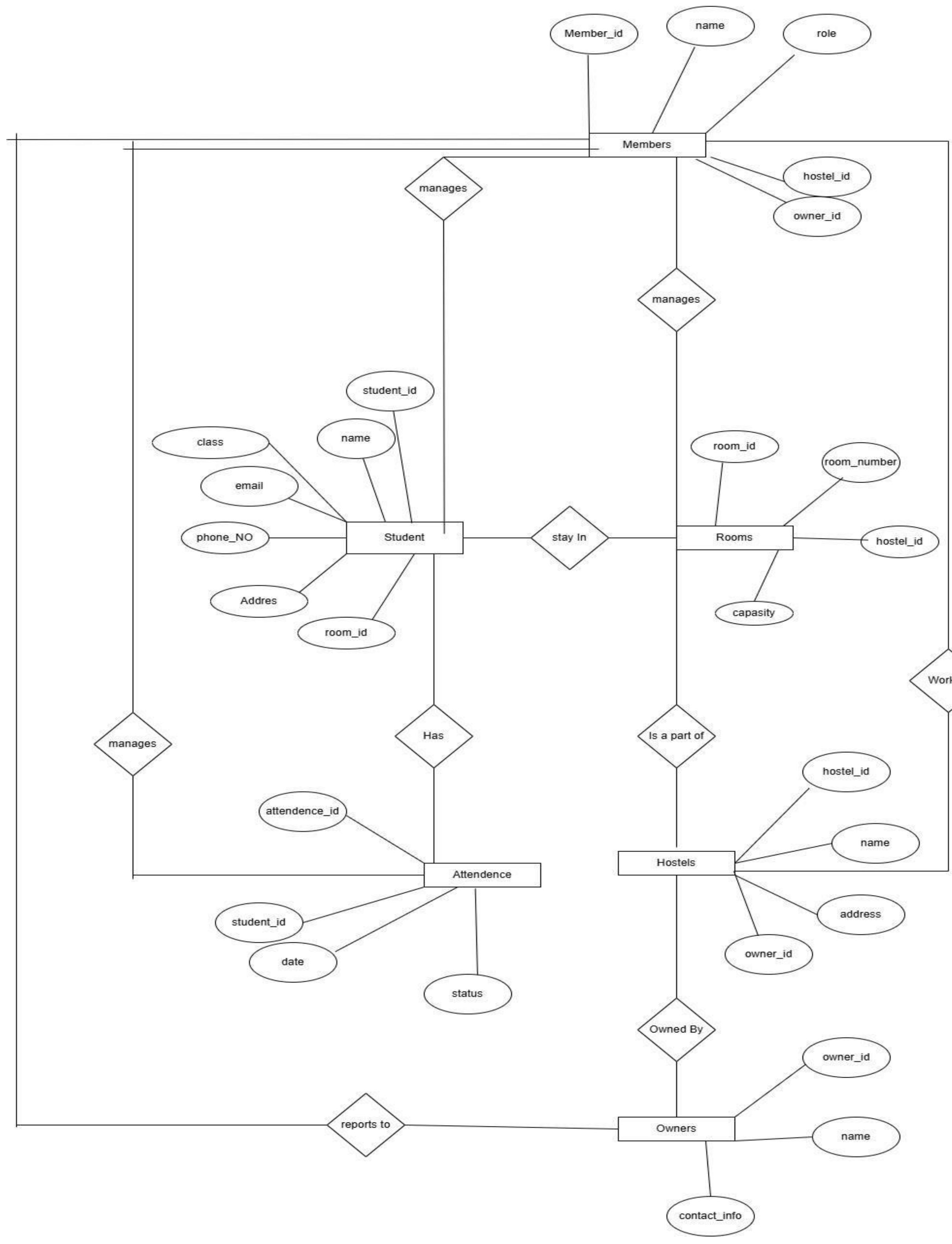
Each hostel is managed by designated owners and employs multiple members, such as wardens, maintenance staff, and administrative personnel. Hostels consist of various rooms, classified by attributes such as room type (e.g., 4-bed, 8-bed) and capacity. Students are allocated to specific rooms, and the system tracks this allocation, ensuring no room exceeds its maximum capacity.

Hostels are geographically situated within the university campus and linked to their owners, who oversee multiple hostels. Members are assigned roles and responsibilities in managing students, rooms, and general operations.

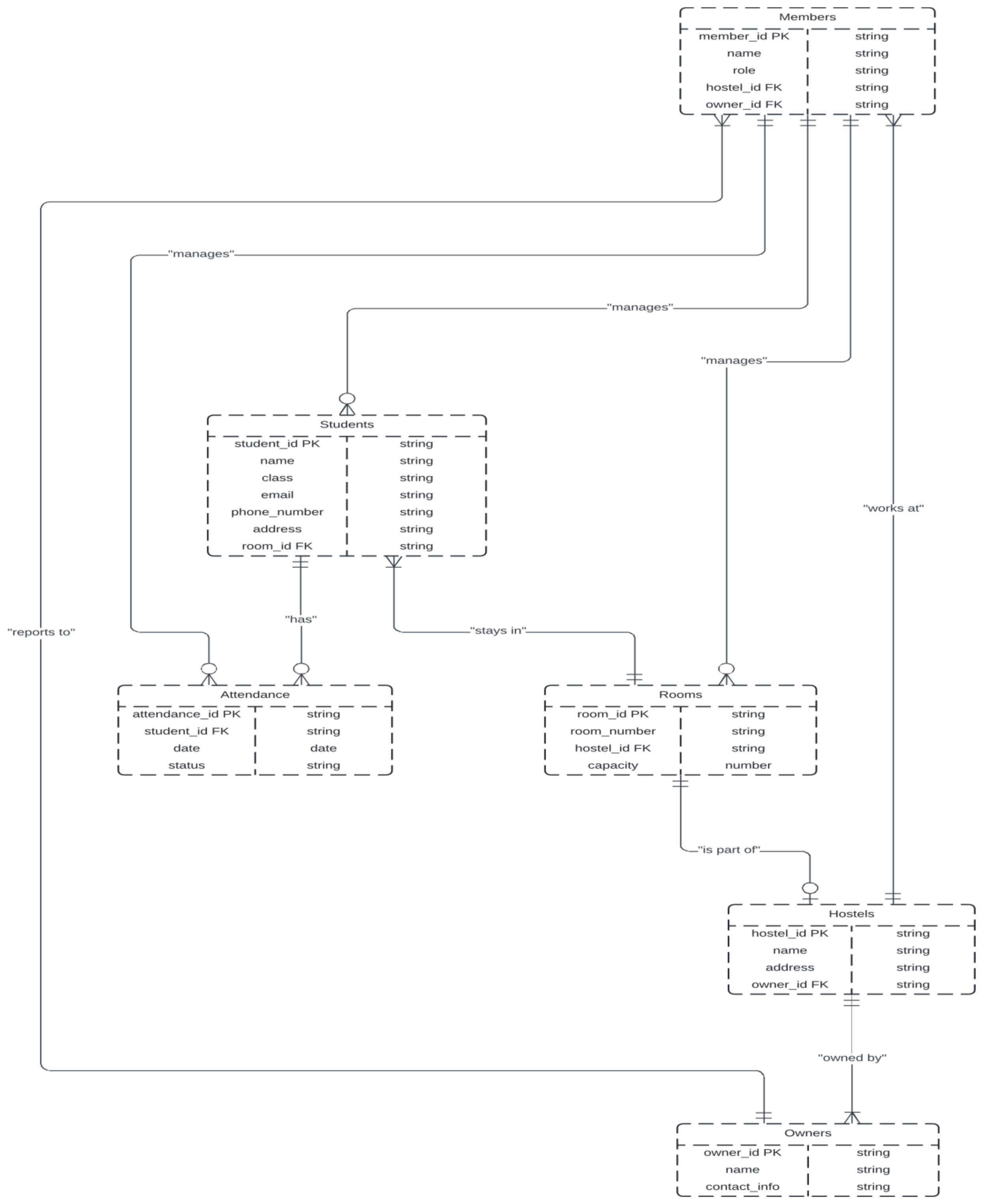
Attendance records are maintained daily for each student, logging their presence or absence.

The system also establishes relationships between owners, members, students, rooms, and hostels, ensuring efficient communication and accountability. Additionally, a comprehensive history of room allocations, student attendance, and administrative activities is maintained to support effective management and future analysis.

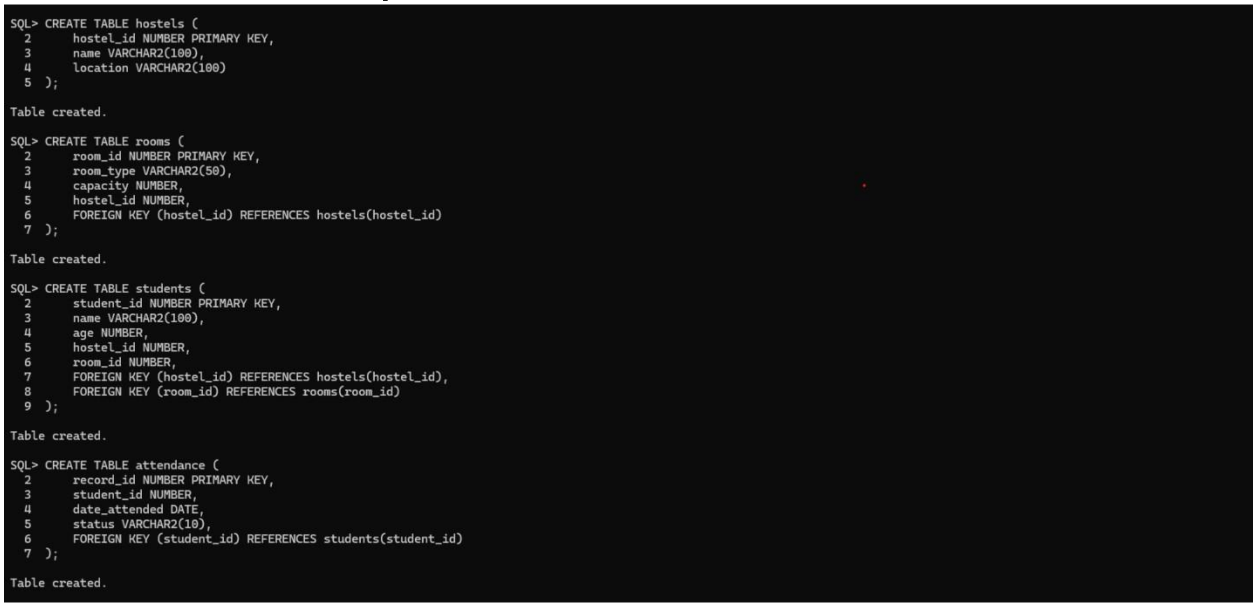
ERD:


Member_id
name
role
Members
hostel_id
owner_id
manages
manages
student_id
class
name
email
phone_NO
Student
Addres
room_id
stay In
room_id
room_number
Rooms
hostel_id
capasity
Work at
manages
Has
Is a part of
hostel_id
attendence_id
name
Hostels
Attendence
student_id
address
date
owner_id
status
Owned By
owner_id
reports to
Owners
name
contact_info

Conceptual Schema:

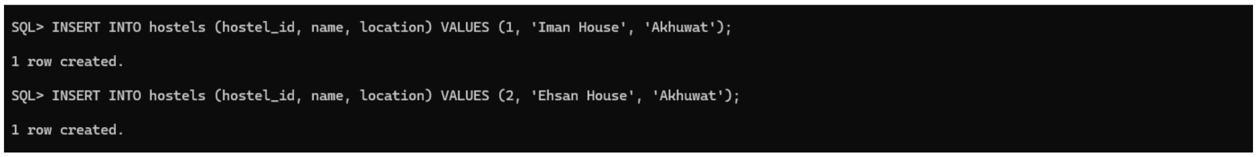


2. **Database creation and implementation of tables.**

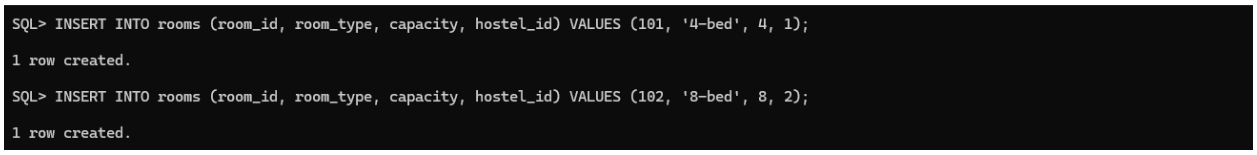


```
SQL> CREATE TABLE hostels (
      hostel_id NUMBER PRIMARY KEY,
      name VARCHAR2(100),
      location VARCHAR2(100)
  );

Table created.

SQL> CREATE TABLE rooms (
      room_id NUMBER PRIMARY KEY,
      room_type VARCHAR2(50),
      capacity NUMBER,
      hostel_id NUMBER,
      FOREIGN KEY (hostel_id) REFERENCES hostels(hostel_id)
  );

Table created.

SQL> CREATE TABLE students (
      student_id NUMBER PRIMARY KEY,
      name VARCHAR2(100),
      age NUMBER,
      hostel_id NUMBER,
      room_id NUMBER,
      FOREIGN KEY (hostel_id) REFERENCES hostels(hostel_id),
      FOREIGN KEY (room_id) REFERENCES rooms(room_id)
  );

Table created.

SQL> CREATE TABLE attendance (
      record_id NUMBER PRIMARY KEY,
      student_id NUMBER,
      date_attended DATE,
      status VARCHAR2(10),
      FOREIGN KEY (student_id) REFERENCES students(student_id)
  );

Table created.
```

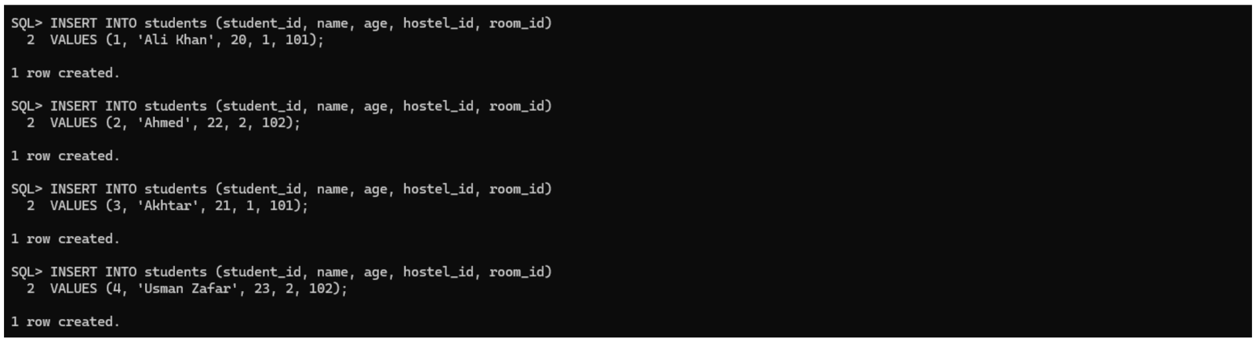
3. **Basic CRUD operations performed on the tables.**



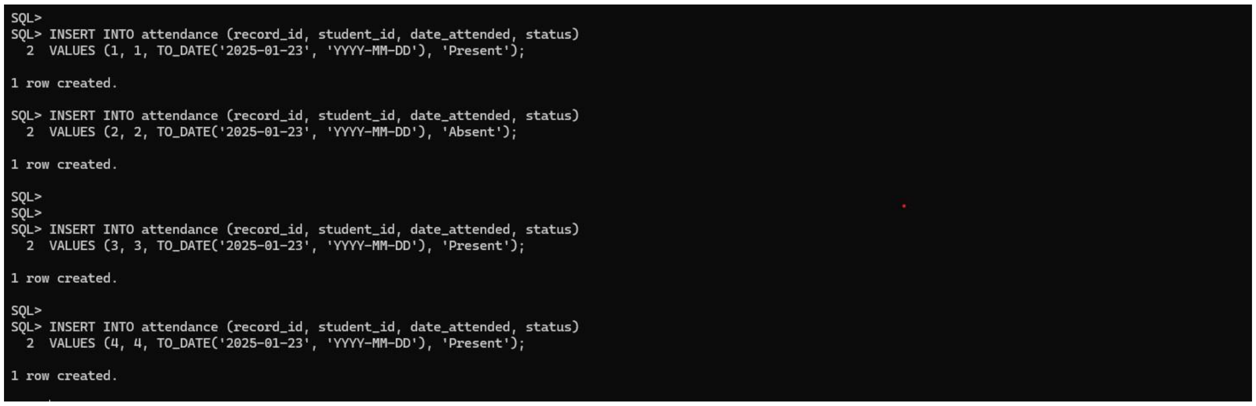
```
SQL> INSERT INTO hostels (hostel_id, name, location) VALUES (1, 'Iman House', 'Akhuwat');

1 row created.

SQL> INSERT INTO hostels (hostel_id, name, location) VALUES (2, 'Ehsan House', 'Akhuwat');

1 row created.
```



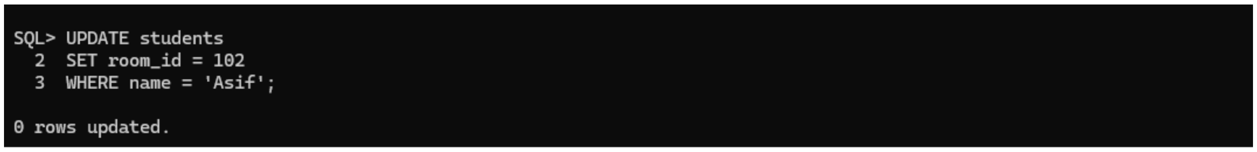
```
SQL> INSERT INTO rooms (room_id, room_type, capacity, hostel_id) VALUES (101, '4-bed', 4, 1);

1 row created.

SQL> INSERT INTO rooms (room_id, room_type, capacity, hostel_id) VALUES (102, '8-bed', 8, 2);

1 row created.
```



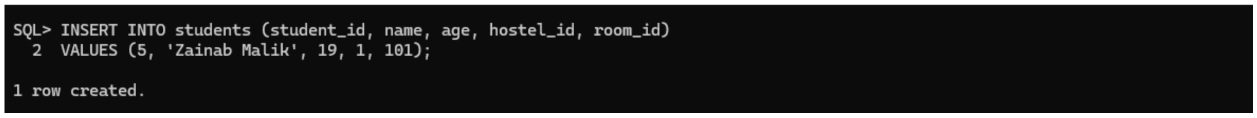
```
SQL> INSERT INTO students (student_id, name, age, hostel_id, room_id)
  2  VALUES (1, 'Ali Khan', 20, 1, 101);

1 row created.

SQL> INSERT INTO students (student_id, name, age, hostel_id, room_id)
  2  VALUES (2, 'Ahmed', 22, 2, 102);

1 row created.

SQL> INSERT INTO students (student_id, name, age, hostel_id, room_id)
  2  VALUES (3, 'Akhtar', 21, 1, 101);

1 row created.

SQL> INSERT INTO students (student_id, name, age, hostel_id, room_id)
  2  VALUES (4, 'Usman Zafar', 23, 2, 102);

1 row created.
```



```
SQL>
SQL> INSERT INTO attendance (record_id, student_id, date_attended, status)
  2  VALUES (1, 1, TO_DATE('2025-01-23', 'YYYY-MM-DD'), 'Present');

1 row created.

SQL> INSERT INTO attendance (record_id, student_id, date_attended, status)
  2  VALUES (2, 2, TO_DATE('2025-01-23', 'YYYY-MM-DD'), 'Absent');

1 row created.

SQL>
SQL>
SQL> INSERT INTO attendance (record_id, student_id, date_attended, status)
  2  VALUES (3, 3, TO_DATE('2025-01-23', 'YYYY-MM-DD'), 'Present');

1 row created.

SQL>
SQL> INSERT INTO attendance (record_id, student_id, date_attended, status)
  2  VALUES (4, 4, TO_DATE('2025-01-23', 'YYYY-MM-DD'), 'Present');

1 row created.
```



```
SQL> UPDATE students
  2  SET room_id = 102
  3  WHERE name = 'Asif';

0 rows updated.
```



```
SQL> INSERT INTO students (student_id, name, age, hostel_id, room_id)
  2  VALUES (5, 'Zainab Malik', 19, 1, 101);

1 row created.
```

Week 4: Tablespaces and Undo Tablespaces

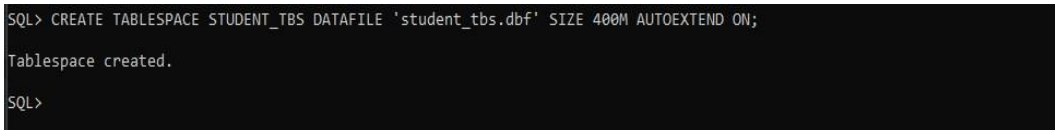
Focus Topics: 1. Creating tablespaces.

2. **Managing tablespaces (extending size, adding data files).**
3. **Creating and managing undo tablespaces.**

1. **Create Tablespaces**:

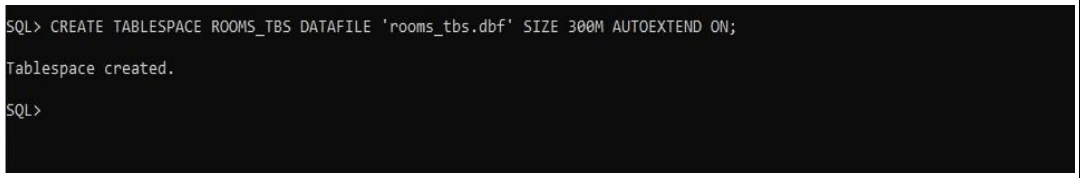
Create tablespaces for different modules:

- STUDENT_TBS: For student information and attendance.



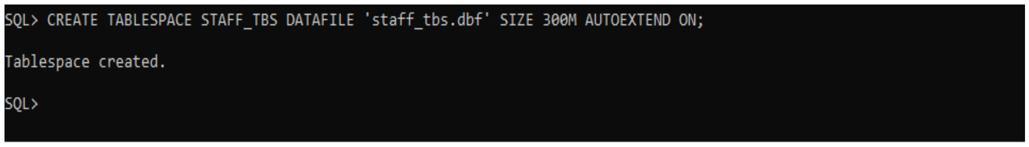
```
SQL> CREATE TABLESPACE STUDENT_TBS DATAFILE 'student_tbs.dbf' SIZE 400M AUTOEXTEND ON;

Tablespace created.

SQL>
```

- ROOMS_TBS: For room and allocation details.



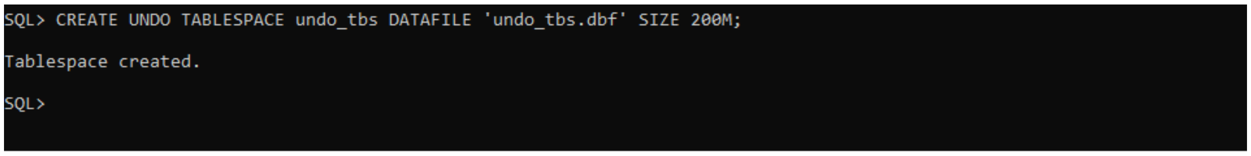
```
SQL> CREATE TABLESPACE ROOMS_TBS DATAFILE 'rooms_tbs.dbf' SIZE 300M AUTOEXTEND ON;

Tablespace created.

SQL>
```

- STAFF_TBS: For hostel staff and roles.



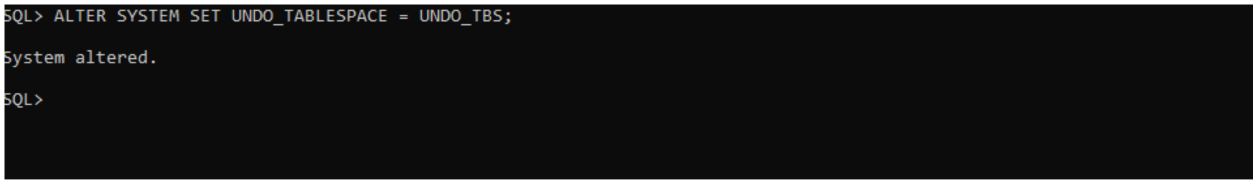
```
SQL> CREATE TABLESPACE STAFF_TBS DATAFILE 'staff_tbs.dbf' SIZE 300M AUTOEXTEND ON;

Tablespace created.

SQL>
```

2. **Managing tablespaces (extending size, adding data files).**



```
SQL> CREATE UNDO TABLESPACE undo_tbs DATAFILE 'undo_tbs.dbf' SIZE 200M;

Tablespace created.

SQL>
```

3. **Creating and managing undo tablespaces.**



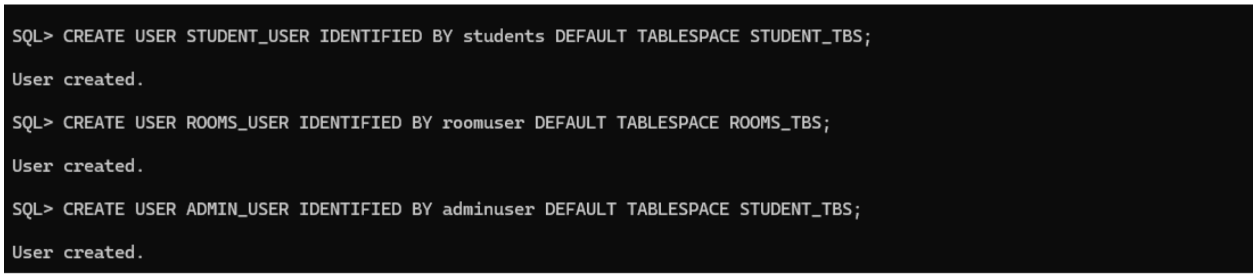
```
SQL> ALTER SYSTEM SET UNDO_TABLESPACE = UNDO_TBS;

System altered.

SQL>
```

Week 5: User Management and Quotas Focus

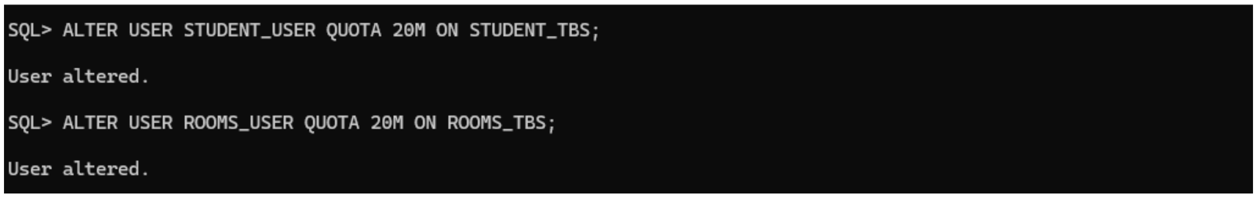
Topics:

1. **Creating users with password expiry.**
2. **Assigning quotas to users.**
3. **Dropping users.**

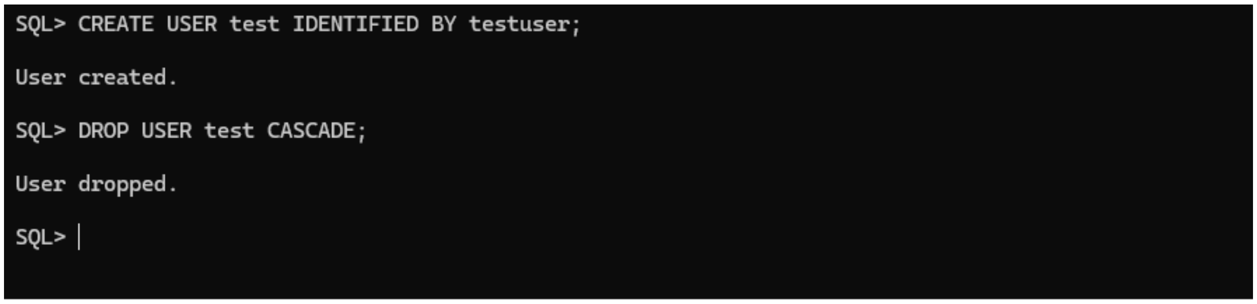
1. **Creating users with password expiry.**


```
SQL> CREATE USER STUDENT_USER IDENTIFIED BY students DEFAULT TABLESPACE STUDENT_TBS;

User created.

SQL> CREATE USER ROOMS_USER IDENTIFIED BY roomuser DEFAULT TABLESPACE ROOMS_TBS;

User created.

SQL> CREATE USER ADMIN_USER IDENTIFIED BY adminuser DEFAULT TABLESPACE STUDENT_TBS;

User created.
```

2. **Assigning quotas to users.**


```
SQL> ALTER USER STUDENT_USER QUOTA 20M ON STUDENT_TBS;

User altered.

SQL> ALTER USER ROOMS_USER QUOTA 20M ON ROOMS_TBS;

User altered.
```

3. **Dropping users.**


```
SQL> CREATE USER test IDENTIFIED BY testuser;

User created.

SQL> DROP USER test CASCADE;

User dropped.

SQL>
```

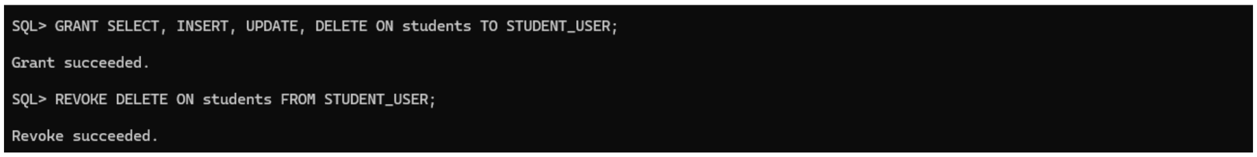
Week 6: Privileges, Roles, Profiles, and Indexes Focus

Topics:

1. **Granting and revoking user privileges.**
2. **Creating roles and profiles.**

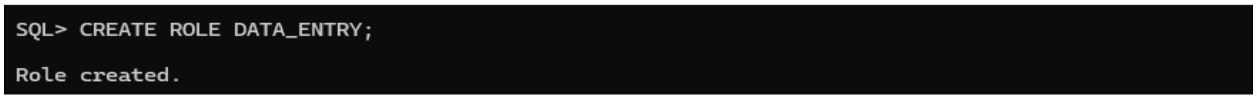
3. **Assigning roles to users.**
4. **Creating and managing indexes.**

1. **Granting and revoking user privileges.**

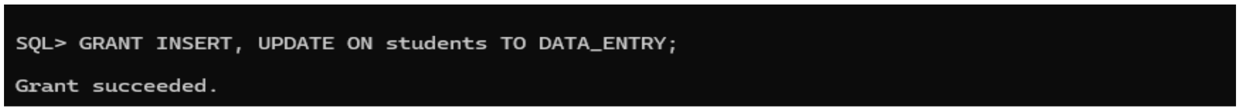


```
SQL> GRANT SELECT, INSERT, UPDATE, DELETE ON students TO STUDENT_USER;

Grant succeeded.

SQL> REVOKE DELETE ON students FROM STUDENT_USER;

Revoke succeeded.
```

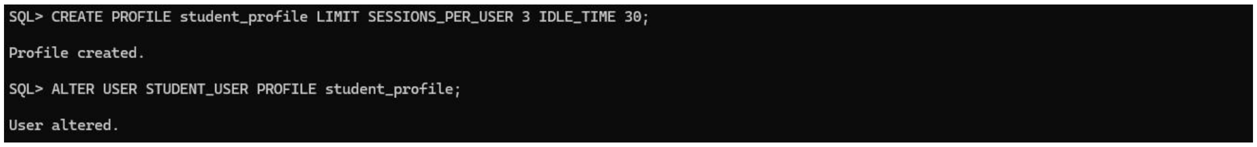
2. **Creating roles and profiles.**



```
SQL> CREATE ROLE DATA_ENTRY;

Role created.
```

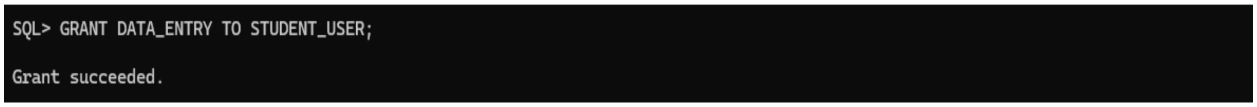


```
SQL> GRANT INSERT, UPDATE ON students TO DATA_ENTRY;

Grant succeeded.
```



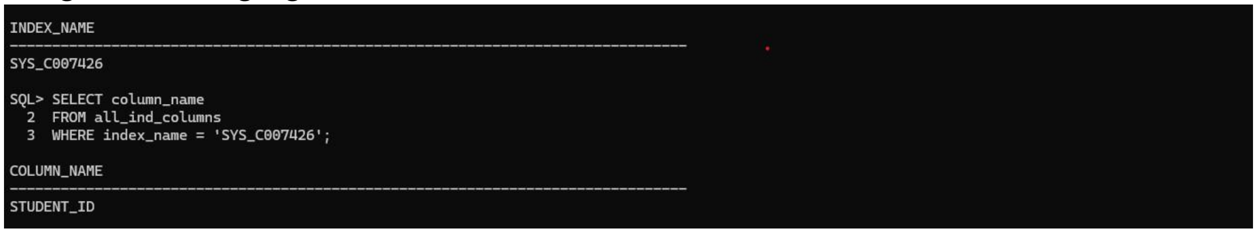
```
SQL> CREATE PROFILE student_profile LIMIT SESSIONS_PER_USER 3 IDLE_TIME 30;

Profile created.

SQL> ALTER USER STUDENT_USER PROFILE student_profile;

User altered.
```

3. **Assigning roles to users.**



```
SQL> GRANT DATA_ENTRY TO STUDENT_USER;

Grant succeeded.
```

4. **Creating and managing indexes.**



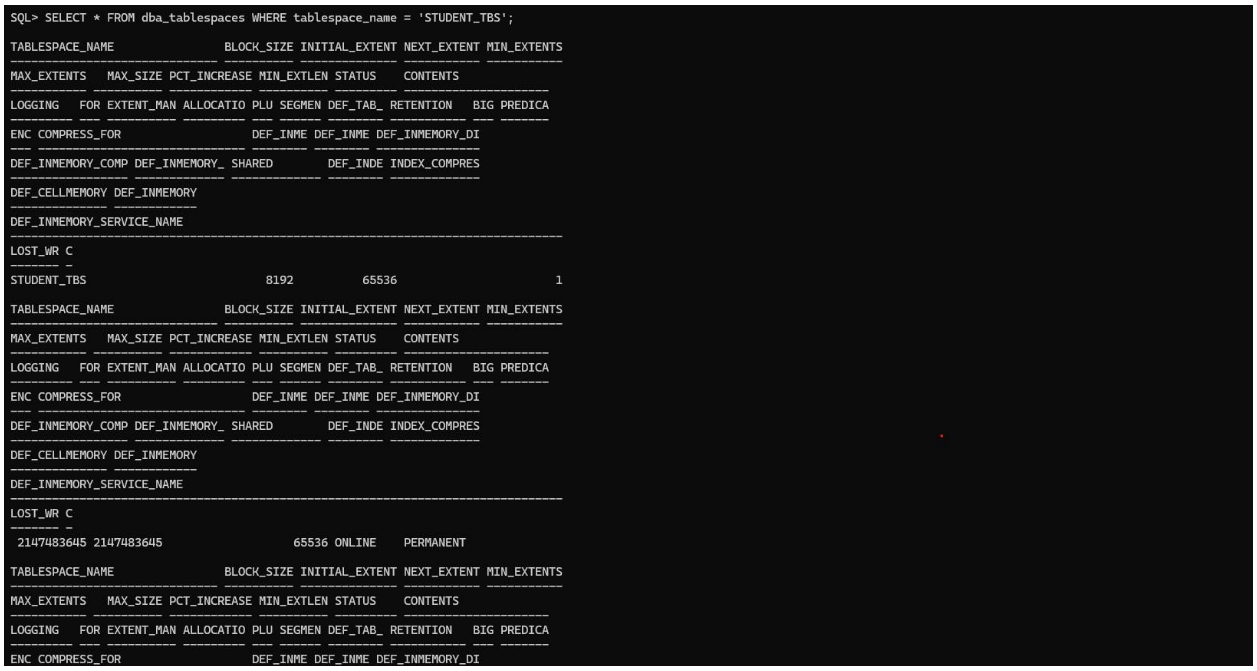
```
INDEX_NAME
--------------------------------------------------------------------------------
SYS_C007426

SQL> SELECT column_name
  2  FROM all_ind_columns
  3  WHERE index_name = 'SYS_C007426';

COLUMN_NAME
--------------------------------------------------------------------------------
STUDENT_ID
```

Week 7: Data Dictionary, Flashback, and Recovery

Focus Topics:

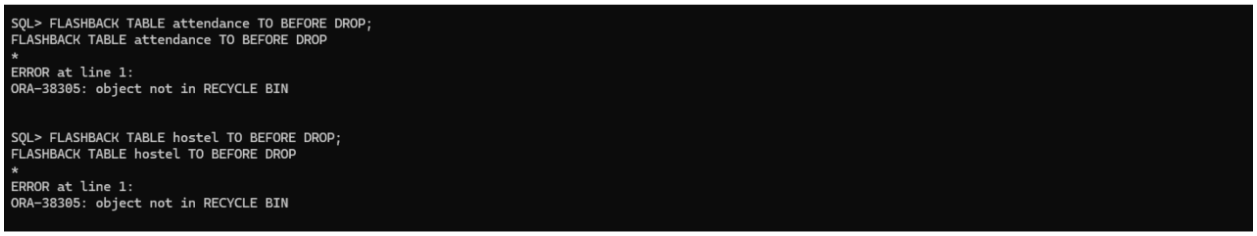
1. **Using the Data Dictionary.**
2. **Flashback operations.**
3. **Recovery using RMAN.**

1. Using the Data Dictionary.



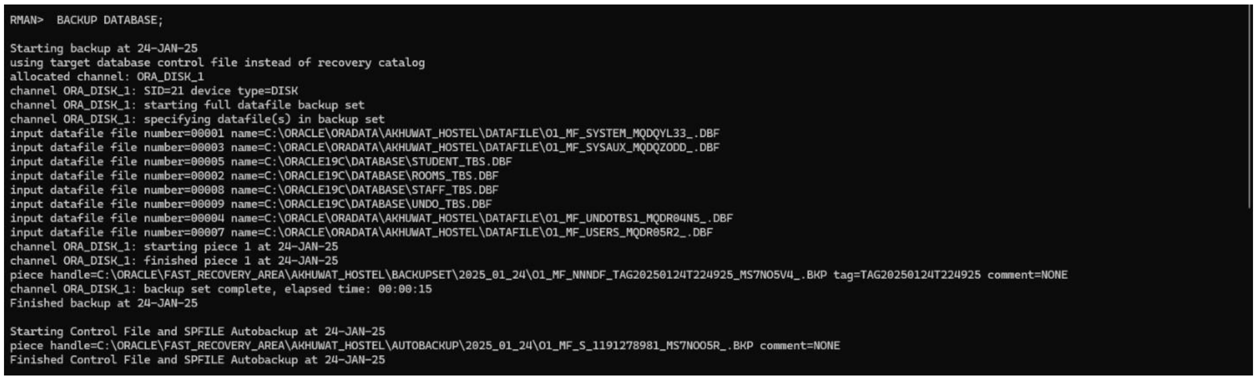
```
SQL> SELECT * FROM dba_tablespaces WHERE tablespace_name = 'STUDENT_TBS';

TABLESPACE_NAME                BLOCK_SIZE INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS
------------------------------ ---------- -------------- ----------- -----------
MAX_EXTENTS   MAX_SIZE PCT_INCREASE MIN_EXTLEN STATUS    CONTENTS
----------- ---------- ------------ ---------- --------- ---------------------
LOGGING   FOR EXTENT_MAN ALLOCATIO PLU SEGMEN DEF_TAB_ RETENTION   BIG PREDICA
--------- --- ---------- --------- --- ------ -------- ----------- --- -------
ENC COMPRESS_FOR                   DEF_INME DEF_INME DEF_INMEMORY_DI
--- ------------------------------ -------- -------- ---------------
DEF_INMEMORY_COMP DEF_INMEMORY_ SHARED        DEF_INDE INDEX_COMPRES
----------------- ------------- ------------- -------- -------------
DEF_CELLMEMORY DEF_INMEMORY
-------------- ------------
DEF_INMEMORY_SERVICE_NAME
--------------------------------------------------------------------------------
LOST_WR C
------- -
STUDENT_TBS                          8192          65536                       1

TABLESPACE_NAME                BLOCK_SIZE INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS
------------------------------ ---------- -------------- ----------- -----------
MAX_EXTENTS   MAX_SIZE PCT_INCREASE MIN_EXTLEN STATUS    CONTENTS
----------- ---------- ------------ ---------- --------- ---------------------
LOGGING   FOR EXTENT_MAN ALLOCATIO PLU SEGMEN DEF_TAB_ RETENTION   BIG PREDICA
--------- --- ---------- --------- --- ------ -------- ----------- --- -------
ENC COMPRESS_FOR                   DEF_INME DEF_INME DEF_INMEMORY_DI
--- ------------------------------ -------- -------- ---------------
DEF_INMEMORY_COMP DEF_INMEMORY_ SHARED        DEF_INDE INDEX_COMPRES
----------------- ------------- ------------- -------- -------------
DEF_CELLMEMORY DEF_INMEMORY
-------------- ------------
DEF_INMEMORY_SERVICE_NAME
--------------------------------------------------------------------------------
LOST_WR C
------- -
 2147483645 2147483645                   65536 ONLINE    PERMANENT

TABLESPACE_NAME                BLOCK_SIZE INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS
------------------------------ ---------- -------------- ----------- -----------
MAX_EXTENTS   MAX_SIZE PCT_INCREASE MIN_EXTLEN STATUS    CONTENTS
----------- ---------- ------------ ---------- --------- ---------------------
LOGGING   FOR EXTENT_MAN ALLOCATIO PLU SEGMEN DEF_TAB_ RETENTION   BIG PREDICA
--------- --- ---------- --------- --- ------ -------- ----------- --- -------
ENC COMPRESS_FOR                   DEF_INME DEF_INME DEF_INMEMORY_DI
```

2. Flashback operations.

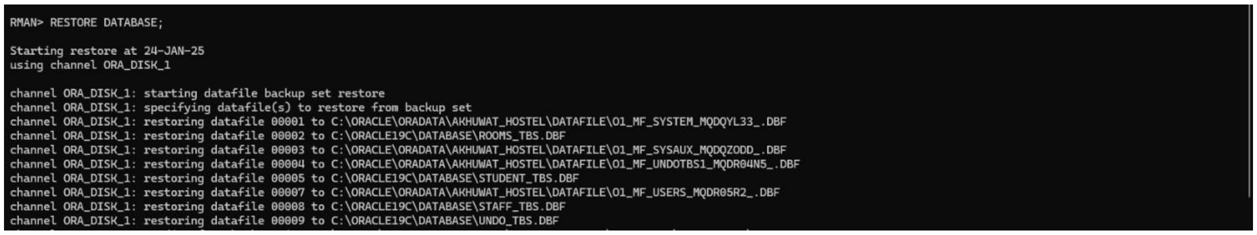


```
SQL> FLASHBACK TABLE attendance TO BEFORE DROP;
FLASHBACK TABLE attendance TO BEFORE DROP
*
ERROR at line 1:
ORA-38305: object not in RECYCLE BIN


SQL> FLASHBACK TABLE hostel TO BEFORE DROP;
FLASHBACK TABLE hostel TO BEFORE DROP
*
ERROR at line 1:
ORA-38305: object not in RECYCLE BIN
```

3. Recovery using RMAN.



```
RMAN>  BACKUP DATABASE;

Starting backup at 24-JAN-25
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=21 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001 name=C:\ORACLE\ORADATA\AKHUWAT_HOSTEL\DATAFILE\O1_MF_SYSTEM_MQDQYL33_.DBF
input datafile file number=00003 name=C:\ORACLE\ORADATA\AKHUWAT_HOSTEL\DATAFILE\O1_MF_SYSAUX_MQDQZODD_.DBF
input datafile file number=00005 name=C:\ORACLE19C\DATABASE\STUDENT_TBS.DBF
input datafile file number=00002 name=C:\ORACLE19C\DATABASE\ROOMS_TBS.DBF
input datafile file number=00008 name=C:\ORACLE19C\DATABASE\STAFF_TBS.DBF
input datafile file number=00009 name=C:\ORACLE19C\DATABASE\UNDO_TBS.DBF
input datafile file number=00004 name=C:\ORACLE\ORADATA\AKHUWAT_HOSTEL\DATAFILE\O1_MF_UNDOTBS1_MQDR04N5_.DBF
input datafile file number=00007 name=C:\ORACLE\ORADATA\AKHUWAT_HOSTEL\DATAFILE\O1_MF_USERS_MQDR05R2_.DBF
channel ORA_DISK_1: starting piece 1 at 24-JAN-25
channel ORA_DISK_1: finished piece 1 at 24-JAN-25
piece handle=C:\ORACLE\FAST_RECOVERY_AREA\AKHUWAT_HOSTEL\BACKUPSET\2025_01_24\O1_MF_NNNDF_TAG20250124T224925_MS7NO5V4_.BKP tag=TAG20250124T224925 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:15
Finished backup at 24-JAN-25

Starting Control File and SPFILE Autobackup at 24-JAN-25
piece handle=C:\ORACLE\FAST_RECOVERY_AREA\AKHUWAT_HOSTEL\AUTOBACKUP\2025_01_24\O1_MF_S_1191278981_MS7NOO5R_.BKP comment=NONE
Finished Control File and SPFILE Autobackup at 24-JAN-25
```



```
RMAN> RESTORE DATABASE;

Starting restore at 24-JAN-25
using channel ORA_DISK_1

channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
channel ORA_DISK_1: restoring datafile 00001 to C:\ORACLE\ORADATA\AKHUWAT_HOSTEL\DATAFILE\O1_MF_SYSTEM_MQDQYL33_.DBF
channel ORA_DISK_1: restoring datafile 00002 to C:\ORACLE19C\DATABASE\ROOMS_TBS.DBF
channel ORA_DISK_1: restoring datafile 00003 to C:\ORACLE\ORADATA\AKHUWAT_HOSTEL\DATAFILE\O1_MF_SYSAUX_MQDQZODD_.DBF
channel ORA_DISK_1: restoring datafile 00004 to C:\ORACLE\ORADATA\AKHUWAT_HOSTEL\DATAFILE\O1_MF_UNDOTBS1_MQDR04N5_.DBF
channel ORA_DISK_1: restoring datafile 00005 to C:\ORACLE19C\DATABASE\STUDENT_TBS.DBF
channel ORA_DISK_1: restoring datafile 00007 to C:\ORACLE\ORADATA\AKHUWAT_HOSTEL\DATAFILE\O1_MF_USERS_MQDR05R2_.DBF
channel ORA_DISK_1: restoring datafile 00008 to C:\ORACLE19C\DATABASE\STAFF_TBS.DBF
channel ORA_DISK_1: restoring datafile 00009 to C:\ORACLE19C\DATABASE\UNDO_TBS.DBF
```