

CN5000 Tutorials

u2613621

Jayrup Nakawala

Table of contents

1 Week 1	5
1.1 Compare and contrast between the main characteristics of the database and file-based approach.	5
1.2 Describe four government sectors that would be expected to make use of database systems.	5
1.3 What is meant by the terms:	5
1.4 Describe the role of database management systems (DBMS) in the database approach. Discuss why knowledge of DBMS is important for database administrators.	6
1.5 Describe the five components of the DBMS environment.	6
1.6 What are the advantages of database management systems? Are there any potential disadvantages?	7
1.7 Discuss the roles of the following personnel in the database environment:	7
1.8 Discuss the three generations of DBMSs.	8
1.9 Why are views an important aspect of database management systems?	8
2 Week 2	9
2.1 Describe the concept of database schema and explain the three types of schema in a database.	9
2.2 What is a data model? What are the three main components of a data model?	9
2.3 What functions and services would you expect a multi-user DBMS provide?	9
2.4 Which functions and services would not be required for a standalone PC DBMS?	10
2.5 Describe the function a system catalogue. What are the benefits of having a system catalog?	10
2.6 What is the difference between DDL and DML?	10
2.7 Name three record-based data models. Discuss the main differences between these data models.	10
3 Week 3	11
3.1 Capture the screenshots of your solution, using snipping tool and put them into your document.	11
3.2 What are the main stages of the database system development lifecycle? Depict the stages using www.draw.io . For each stage, state if it is mandatory or optional.	12
3.3 Discuss the main approaches to database design. Discuss the contexts where each is appropriate.	13
3.4 What are the three phases of database design? How are they related to each other?	14

3.5 The following are problems that have been identified during the testing process in the development of a new system. In which part of the life cycle do you think these problem could have originated and been identified by a thorough review following that stage in the development life cycle?	14
4 week 4	15
4.1 For the case-study design of Company, design the full (Advance) E-R diagram that shows the link between them and list all the attributes along with PK and FK using oracle data modeler. . .	15
4.2 Which normal form (Normalization) do the attributes in these entities belong to ?	16
4.3 What will be the 0NF attributes of this system ?	16
4.4 Draw a simple usecase for the above system.	17
5 Week 5	18
5.1 Output for SQL Command:	18
5.2 Output for SQL Command:	18
6 Week 6	19
6.1 Create tables and insert values in them based on the given erd. .	19
6.2 Use basic select statements to provide rows of each table and provide screenshot.	21
7 Week 7	23
7.1 Queries	23
7.2 Creating Views	35
7.3 Creating and Executing Procedures	36
7.4 Creating and Executing Functions	39
8 Week 8	43
8.1 Consider the COMPANY database provided in the previous assignments. Using the syntax of Oracle triggers, create triggers to do the following:	43
9 Week 9	46
9.1 What is a Transaction in the context of Databases?	46
9.2 What is ACID in Transaction? Define each identified.	46
9.3 What are the different states of a transaction?	46
9.4 Define COMMIT and ROLLBACK and give example of each. .	46
9.5 What are the three main recovery techniques?	47
10 Week 10	48
10.1 What is the purpose of database security in general?	48
10.2 Describe the approaches for securing DBMSs on the Web.	48
10.3 Various threats exist on computer system, describe the possible threats and the likelihood of occurring for the followings:	48

10.4 What is the purpose of access control matrix? Illustrate your answer with an example.	49
10.5 Explain how views can be used to improve security of DBMS. . .	49
10.6 What is difference between Discretionary Access Control (DAC) and Mandatory Access Control (MAC)	49

1 Week 1

1.1 Compare and contrast between the main characteristics of the database and file-based approach.

- **File-based Approach:** A file based system is a collection of applications which uses their own, separate databases to manage their data. Resulting in data redundancy and lower reliability. Although it is easier to set up. This approach allows data to be stored in a structured or unstructured way, eg, csv, json, docx, xlss.
- **Database Approach:** A database management system, creates a system for the collection of application which allows them to store data centrally. This approach allows the user to create relationships between data through primary and foreign keys and helps increase data integrity through constraints and views. This approach allows data to be stored only in a structured way.

1.2 Describe four government sectors that would be expected to make use of database systems.

- **Sector for Transport Management** – for traffic management, drivers and vehicle records, transport planning, etc.
- **Sector for Education** – national pupil database, exams and assessment records, school census results, etc.
- **Sector for Business & Trade** – business registration records, supply chain management, international trade, etc.
- **Sector for Health & Social Care** – patient records, public health databases, social care, etc.

1.3 What is meant by the terms:

- **Database:** In simplest terms, it is a collection of structured related data and its descriptions. Which is designed to meet the information needs of an organization.
- **Database Management System:** A management system which allows the user to interact with the database. Which includes the ability to create, define, maintain and control access to the database via the Data Manipulation Language. It serves as an intermediary between the database and the application (or users). Examples: MySQL, MongoDB, PostgreSQL, etc.
- **Database Application Program:** It is an application designed to interact with a database in a more user-friendly interface making the database interactions easier for non-technical users. Some common examples include: Microsoft Access, Oracle forms, etc.
- **Data Independence:** Data independence refers to the way a DBMS separates the logical structure of data from its physical representation.

ates the data description from the application, thereby making the applications immune to changes in data descriptions.

- **Security:** Database security is the protection of the database by constricting access to it. For example, the Database Admin can access all the data in the database but, lets say, a manager of the marketing team can only access data referring to their field. The operation allowed can also be restricted, for example, an intern may only be able to view a small subset of data and not be allowed to make any changes to it.
- **Integrity:** Database integrity means the validity and consistency of the data stored which can be expressed in terms of constraints. Constraints are a way of defining rules between a single record or to relationships between records.
- **Views:** A view is a subset of the database. It allows for better security by only letting the people who have access view certain data.

1.4 Describe the role of database management systems (DBMS) in the database approach. Discuss why knowledge of DBMS is important for database administrators.

DBMS are crucial to the database approach. This approach involves organizing, storing, and managing data in a way that is independent of the application programs that use the data. Database Administrators are responsible for ensuring that databases operate efficiently, securely, and without disruption. Knowledge of how DBMS works makes them more efficient at their work and allows them to perform tasks much quickly all the while increasing securing and integrity of the data.

1.5 Describe the five components of the DBMS environment.

-Hardware: It refers to the hardware the database runs on. Including, the servers, which run the DBMS, the switches which enable networking and fans which allow for cooling of the systems.

-Software: it refers to the actual DBMS software which handles the data, as well as the OS and any networking software.

-Data: Perhaps the most important part of the DBMS environment. It is the main reason DBMS exists and it also acts as a bridge between the human component and the machine component of the DBMS environment.

-Procedures: These are the rules and instructions which govern the design and use of the database. Including instructions on how to log in, use a specific functionality, start and stop the DBMS, making backups, etc.

-People: There are 4 distinct kinds of people who participate in the DBMS

environment, namely, data and database administrators, database designers, application developers, and end-users.

1.6 What are the advantages of database management systems? Are there any potential disadvantages?

Here are the potential advantages and disadvantages of the database management system:

Advantages	Disadvantages
Control of data redundancy	Complexity
Data consistency	Size of DBMS
More info for the same amount of data	Cost of DBMS
Sharing of data	Cost of conversion
Improved data security	Performance
Enforcement of standards	Additional hardware costs
Improved data integrity	Greater impact of a failure

1.7 Discuss the roles of the following personnel in the database environment:

- **Data Administrator:** is responsible for the management of the data resource, including database planning; development and maintenance of standards, policies and procedures; and conceptual/logical database design.
- **Database Administrator:** is responsible for the physical realization of the database, including physical database design and implementation, security and integrity control, maintenance of the operational system, and ensuring satisfactory performance of the applications for users.
- **Logical Database Designer:** The logical database designer is concerned with identifying the data (that is, the entities and attributes), the relationships between the data, and the constraints on the data that is to be stored in the database.
- **Physical Database Designer:** The physical database designer decides how the logical database design is to be physically realized, including the mapping of logical designs to the sets of tables and constraints, selecting specific storage structure, and security.
- **Application Developer:** Creates the application program that provides the required functionality for the end-users.
- **End-Users:** The end-users are the “clients” of the database, which has been designed and implemented and is being maintained to serve their information needs. They can be novice or sophisticated.

1.8 Discuss the three generations of DBMSs.

The history of Database Management Systems (DBMSs) can be divided into three generations, each representing significant technological advancements and addressing the limitations of the previous generation:

- First-Generation DBMSs: Hierarchical and Network Models
- These ranged from Mid-1960s to early 1970s and had tree based hierarchical structure or complex network models which were cumbersome to maintain and expand.
- Second-Generation DBMSs: Relational Model
- Ranges from 1970s to 1980s, and introduced a fundamentally different relational model which improved data independence.
- Third-Generation DBMSs: Object-Oriented and Object-Relational Models
- From 1990s to present, this generation gave a more object oriented approach allowing to overcome the limitations of the previous generations. This means support for inheritance, encapsulation, and polymorphism and more powerful and efficient tools.

1.9 Why are views an important aspect of database management systems?

A view is, in essence, some subset of the database. It is important for the following reasons:

- Views provide a level of security. Views can be set up to exclude data that some users should not see.
- Views provide a mechanism to customize the appearance of the database.
- A view can present a consistent, unchanging picture of the structure of the database, even if the underlying database is changed

2 Week 2

2.1 Describe the concept of database schema and explain the three types of schema in a database.

For data independence, it is important to keep the database and the description of the database separate. This description of the database is called the database schema. There are three different types of schema based on their level of abstraction.

- External Schema aka, subschema. These are the highest level of abstraction which relates to the different views of the database. It describes the part of the database which is relevant to the user.
- Conceptual Schema This describes what data is stored and its relationships between them, which includes all the entities, attributes and relationships together with integrity constraints.
- Internal Schema This is the lowest level of abstraction.

2.2 What is a data model? What are the three main components of a data model?

A Data Model in Database is a model describing how the real-world objects are organized and how these are associated within the database in a visual representation. It is an abstraction that provides understanding of the database design to the end-users and database designers. It summarizes the description of the database helping to implement the design developed.

Three main components of the data model are: - Structural part that contains the set of rules which the database is structured or built. - Manipulative part constitutes what commands can be used to add, delete or alter the data present in the database. - Integrity constraints that maintain data consistency.

2.3 What functions and services would you expect a multi-user DBMS provide?

A multi-user DBMS should have the following functions and services.

- concurrency
- network connection management
- providing access to all the database servers
- database connection pooling
- legacy database support
- clustering support
- load balancing
- failover

2.4 Which functions and services would not be required for a standalone PC DBMS?

Following are the functions not required for standalone PC DMS:

- Multi-user Access: Since it is a standalone PC DBMS, multi-user access function is not required.
- Network communication: Standalone PC DBMS don't communicate with other computers; network protocols are unnecessary.
- Concurrency Control System: Concurrency Control is needed when multiple users are using and updating the same database simultaneously. Since standalone PC DBMS is accessed by only one user, concurrency control service is not essential.

2.5 Describe the function a system catalogue. What are the benefits of having a system catalog?

Any database management system that describes the database objects has a system catalogue at its core. It includes details on the physical and logical construction of database objects, including packages, tables, views, indexes, and statistical data. System catalogue provides vast information about your database management system. The benefits of having a system catalog include centralizing data, simplifying communications, discovering, and regulating redundancies and inconsistencies, and assuring security and integrity.

2.6 What is the difference between DDL and DML?

- DDL (Data Definition Language): Defines or modifies database structure (e.g., CREATE, ALTER, DROP).
 - Example:

```
CREATE TABLE Students (ID INT, Name VARCHAR(50));
```
- DML (Data Manipulation Language): Manipulates data in the database (e.g., INSERT, UPDATE, DELETE, SELECT).
 - Example:

```
INSERT INTO Students (ID, Name) VALUES (1, 'Alice');
```

2.7 Name three record-based data models. Discuss the main differences between these data models.

Three record-based data models are as follows:

- Hierarchical model: Organizes data in tree-like structure representing data in parent-child hierarchy.
- Network model: Like Hierarchical model Organizes data in graph-like structure representing more complex relationships.
- Relational model: Organizes data in tables – rows and columns. Tables are associated with each other using foreign keys.

3 Week 3

3.1 Capture the screenshots of your solution, using snipping tool and put them into your document.

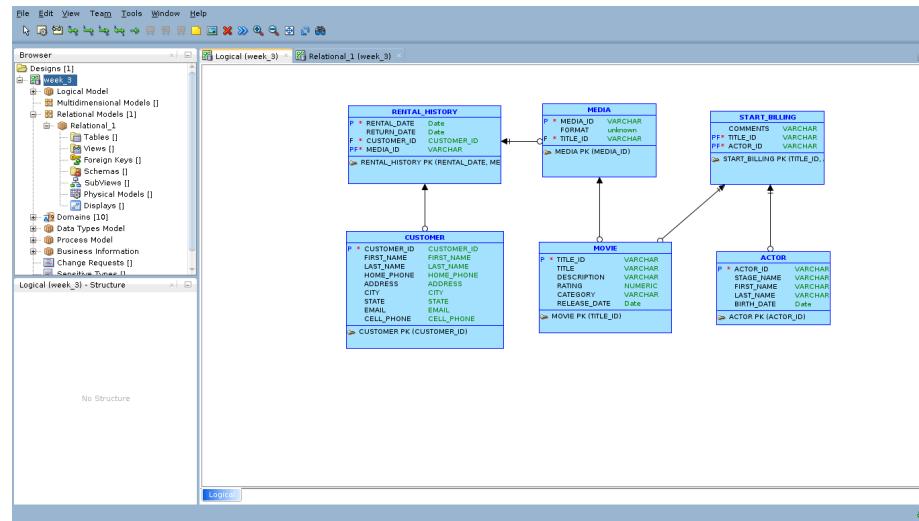


Figure 1: erd

3.2 What are the main stages of the database system development lifecycle? Depict the stages using www.draw.io. For each stage, state if it is mandatory or optional.

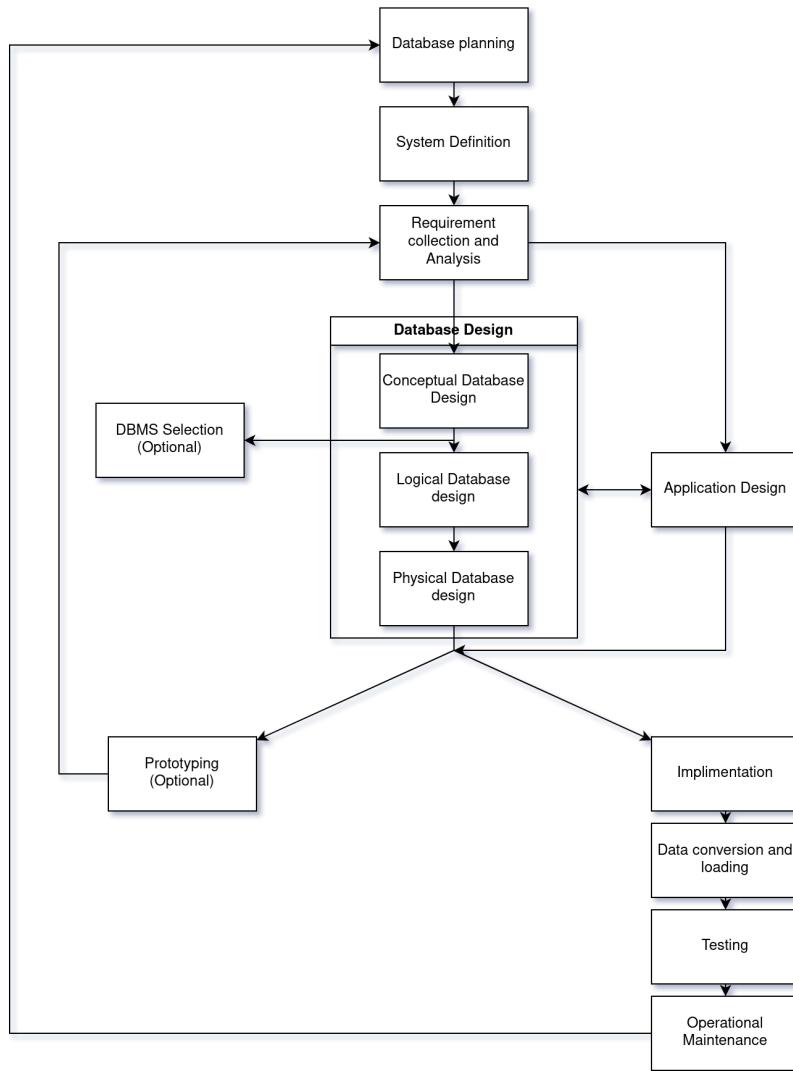


Figure 2: flowchart

Main Stages of the Database Development Lifecycle:

- Database Planning Lays the foundation for all other stages of the project.

(Mandatory)

- System Definition Defines the goals and scope of the project.
- Requirement Collection and Analysis Documents specific needs and assesses project feasibility.
- Database Design Builds the data architecture, including:
 - Conceptual Design
 - Logical Design
 - Physical Design
- DBMS Selection (Optional) Selects a Database Management System (DBMS) that meets project requirements.
- Application Design Plans the use of application programs and user interfaces.
- Prototyping (Optional) Creates a working model of the database.
- Implementation Bridges the gap between idea and execution. (Mandatory)
- Data Conversion and Loading Transfers and uploads existing data into the new system.
- Testing Ensures the system is validated for data integrity through various tests. (Mandatory)

3.3 Discuss the main approaches to database design. Discuss the contexts where each is appropriate.

The main approaches to database design are:

- **Bottom-Up Approach:** This approach starts with the smallest data elements and focuses on their relationships. These elements are gradually combined to form larger entities and structures.
 - Appropriate Context: Ideal for systems where data elements are well-defined, such as legacy systems or projects with detailed existing data but no clear overall structure.
- **Top-Down Approach:** The design begins at a high level by identifying key entities, their attributes, and the relationships between them. The entities are then broken down further to add details.
 - Appropriate Context: Suitable for new systems where an overall structure is clear from the start but specific details will emerge later, like enterprise-level applications.
- **Inside-Out Approach:** This method focuses on defining the most critical or central entities first. The design is then expanded to incorporate related entities and attributes.
 - Appropriate Context: Best for systems where a core functionality or central component is already known, such as modular or specialized applications.
- **Mixed Strategy:** A hybrid approach that combines top-down and bottom-up methods. High-level design is applied to some parts of the system, while detailed data structures are addressed in others.

- Appropriate Context: Useful for complex systems that require flexibility, such as large-scale projects with varying levels of detail across different modules.

3.4 What are the three phases of database design? How are they related to each other?

Conceptual Design

- Defines high-level data requirements using models like Entity-Relationship Diagrams (ERDs).
- Focuses on identifying entities, relationships, and constraints without implementation details.

Logical Design

- Maps the conceptual model into a logical schema compatible with the chosen DBMS.
- Defines tables, attributes, keys, and relationships.

Physical Design

- Implements the logical design by optimizing storage structures, indexing, and access paths for performance.

Relationship : Each phase builds on the previous one: the conceptual design captures what data is needed, the logical design structures it for the DBMS, and the physical design implements and optimizes it for storage and performance.

3.5 The following are problems that have been identified during the testing process in the development of a new system. In which part of the life cycle do you think these problem could have originated and been identified by a thorough review following that stage in the development life cycle?

The performance of the system is poor – failing to respond quickly enough to meet the stated user requirement of interactive, screen-based use.

- Physical Design Phase: The system's performance issues likely stem from poor optimization of storage structures, indexing, or access paths. A thorough review here could have addressed performance requirements.

No backup facilities were included to meet the users' requirement of long-term archival of their data.

- Requirements Collection and Analysis Phase: Backup requirements were likely missed during this phase. Proper documentation and review would have ensured backup and archival needs were addressed.

No user manuals were provided!

- Implementation or Testing Phase: User documentation was overlooked or deprioritized. A thorough review during these stages could have identified and resolved the issue before deployment.

4 week 4

4.1 For the case-study design of Company, design the full (Advance) E-R diagram that shows the link between them and list all the attributes along with PK and FK using oracle data modeler.

- COMPANY
 - Employees, departments, and projects
 - Company is organized into departments
 - Department controls a number of projects
 - Employee: store each employee's name, Social Security number, address, salary, sex (gender), and birth date
 - Keep track of the dependents of each employee

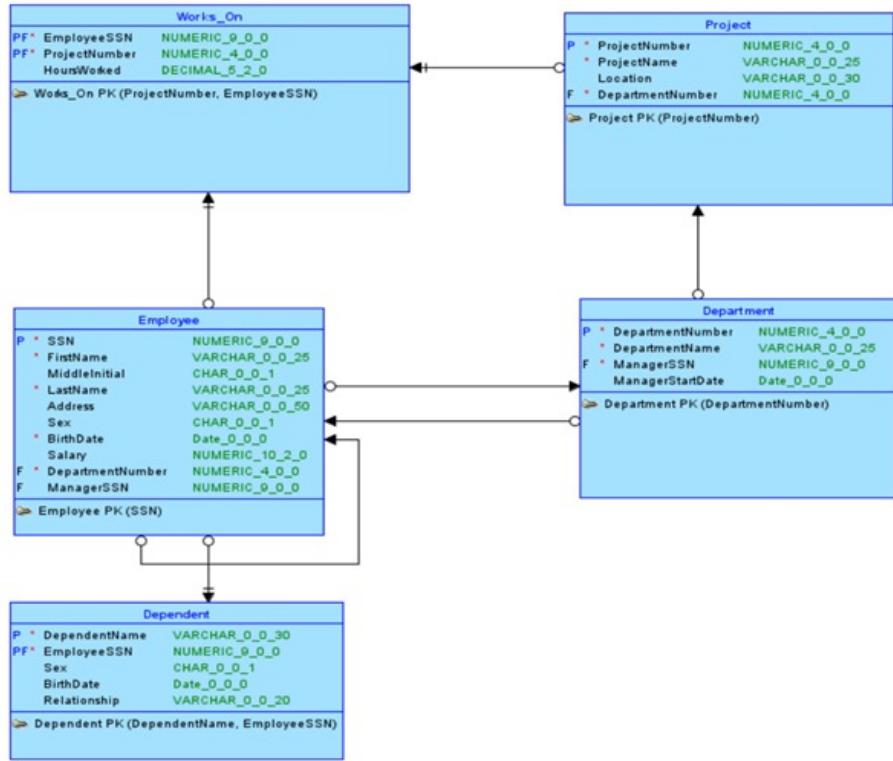


Figure 3: erd

4.2 Which normal form (Normalization) do the attributes in these entities belong to ?

1 NF

4.3 What will be the 0NF attributes of this system ?

As the data is already in 1NF form, there will be no 0NF attributes.

4.4 Draw a simple usecase for the above system.

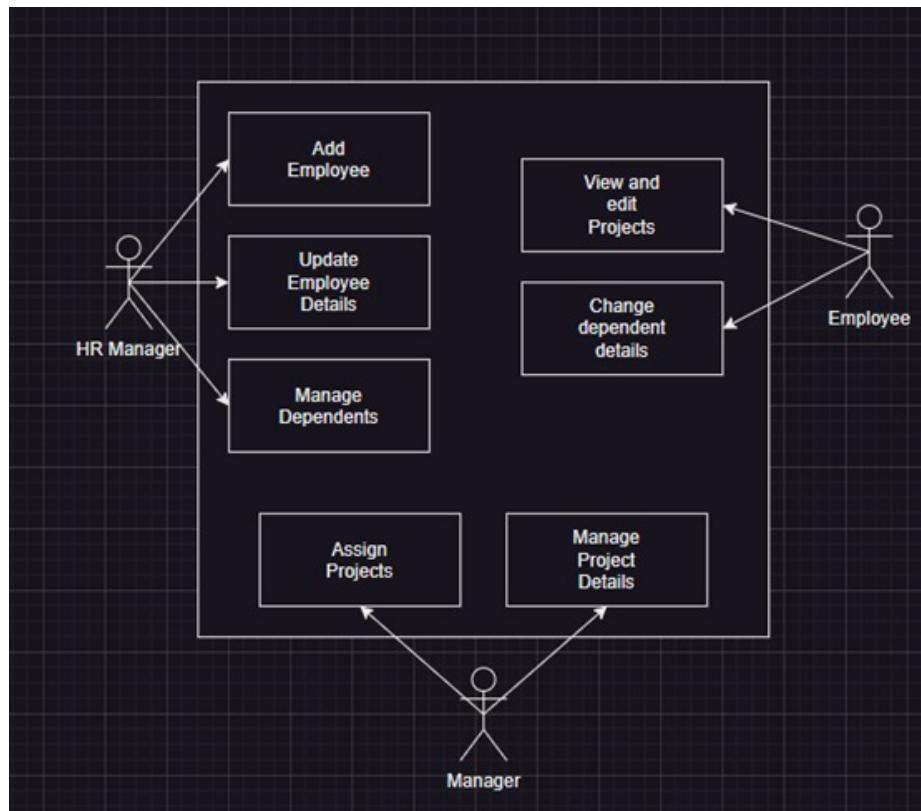


Figure 4: usecase

5 Week 5

5.1 Output for SQL Command:

```
SELECT * FROM EMP
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7300	SMITH	CLERK	7002	12/01/2020	800	-	20
7409	ALLEN	SALESMAN	7698	02/20/2021	1600	300	30
7564	MARTIN	SALESMAN	7698	09/28/2021	1250	1400	30
7698	JLAKE	MANAGER	7819	05/01/2021	2850	-	30
7782	CLARK	MANAGER	7819	06/09/2021	2400	-	10
7839	KING	PRESIDENT	-	11/17/2020	5000	-	10
7844	TURNER	SALESMAN	7308	05/05/2021	900	0	30
7850	ADAMS	CLERK	7788	01/12/2021	1100	-	20
7900	JAMES	CLERK	7698	07/01/2021	950	-	30
7902	FORD	ANALYST	7566	01/05/2021	3000	-	20
7908	MARCUS	MANAGER	7819	04/05/2021	3000	-	10
7988	SCOTT	ANALYST	7566	02/05/2020	3000	-	20
7984	MILLER	CLERK	7782	01/15/2021	1300	-	10
7988	JAPRUD	MANAGER	7819	04/15/2021	3000	-	10
7521	WARD	SALESMAN	7698	02/22/2021	1250	500	30

Figure 5: table1

5.2 Output for SQL Command:

```
SELECT EMPNO, ENAME, JOB_NAME FROM EMP
```

EMPNO	ENAME	JOB
7300	SMITH	CLERK
7409	ALLEN	SALESMAN
7564	WARD	SALESMAN
7566	JONES	MANAGER
7566	MARTIN	SALESMAN
7698	JLAKE	MANAGER
7782	CLARK	MANAGER
7788	SCOTT	ANALYST
7839	KING	PRESIDENT
7844	TURNER	SALESMAN
7850	ADAMS	CLERK
7900	JAMES	CLERK
7902	FORD	ANALYST
7908	MARCUS	MANAGER
7988	SCOTT	ANALYST
7984	MILLER	CLERK
7988	JAPRUD	MANAGER
7521	WARD	SALESMAN

Figure 6: table2

6 Week 6

6.1 Create tables and insert values in them based on the given erd.

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for App Builder, SQL Workshop, Team Development, Gallery, Search, and a user profile for Jayrup Nakawala (u2613621). The main area is titled "SQL Scripts\ Results" and displays the output of a script named "week6". The script status is "Complete" and it took 0.07 seconds to run. The results table has columns: Number, Elapsed, Statement, Feedback, and Rows. The table lists 31 rows of SQL statements and their execution details. Most statements are table creation commands (e.g., create table employee, create table department) which resulted in "Table created." feedback and 0 rows inserted. Rows 7 through 31 are INSERT INTO statements for the "EMPLOYEE" table, each resulting in 1 row(s) inserted. The data for these inserts includes various names and department codes like 'B', 'T', 'A', 'S', 'E', 'K', 'Na', etc.

Number	Elapsed	Statement	Feedback	Rows
1	0.07	create table employee (ssn number(9) primary key, name varchar(25), address varchar(50), phone number(12))	Table created.	0
2	0.03	create table department (dnumber number(2) primary key, dname varchar(14), loc dept_locations(dnumber))	Table created.	0
3	0.02	create table dept_locations (dnumber number(2) , location varchar(15))	Table created.	0
4	0.02	create table project (pnumber number(4) not null, pname varchar(14), dnumber number(2))	Table created.	0
5	0.02	create table work_on (e_ssn number(9) not null, pnumber number(4) not null, hours number(3,1))	Table created.	0
6	0.02	create table dependent (dep_name varchar(25) not null, e_ssn number(9) not null, dnumber number(2))	Table created.	0
7	0.03	INSERT INTO "EMPLOYEE" VALUES (123456789, 'John', 'B', 'Smith')	1 row(s) inserted.	1
8	0.00	INSERT INTO "EMPLOYEE" VALUES (333445555, 'Franklin', 'T', 'Brown')	1 row(s) inserted.	1
9	0.00	INSERT INTO "EMPLOYEE" VALUES (999887777, 'Alicia', 'J', 'Zeeman')	1 row(s) inserted.	1
10	0.00	INSERT INTO "EMPLOYEE" VALUES (987654321, 'Jennifer', 'S', 'King')	1 row(s) inserted.	1
11	0.00	INSERT INTO "EMPLOYEE" VALUES (666884444, 'Ramesh', 'K', 'Narayan')	1 row(s) inserted.	1
12	0.00	INSERT INTO "EMPLOYEE" VALUES (453453453, 'Joyce', 'A', 'Engelmann')	1 row(s) inserted.	1
13	0.00	INSERT INTO "EMPLOYEE" VALUES (987987987, 'Ahmad', 'V', 'Jabber')	1 row(s) inserted.	1
14	0.01	INSERT INTO "EMPLOYEE" VALUES (888665555, 'James', 'E', 'Borg')	1 row(s) inserted.	1
15	0.01	INSERT INTO "DEPARTMENT" VALUES (5, 'Research', 333445555, 'Computer Science')	1 row(s) inserted.	1
16	0.01	INSERT INTO "DEPARTMENT" VALUES (4, 'Administration', 987654321, 'Administration')	1 row(s) inserted.	1
17	0.00	INSERT INTO "DEPARTMENT" VALUES (1, 'Headquarters', 888665555, 'Headquarters')	1 row(s) inserted.	1
18	0.02	INSERT INTO "DEPT_LOCATIONS" VALUES (1, 'Houston')	1 row(s) inserted.	1
19	0.00	INSERT INTO "DEPT_LOCATIONS" VALUES (4, 'Sta rd')	1 row(s) inserted.	1
20	0.00	INSERT INTO "DEPT_LOCATIONS" VALUES (5, 'Bellaire')	1 row(s) inserted.	1
21	0.00	INSERT INTO "DEPT_LOCATIONS" VALUES (5, 'Sugarland')	1 row(s) inserted.	1
22	0.01	INSERT INTO "DEPT_LOCATIONS" VALUES (5, 'Houston')	1 row(s) inserted.	1
23	0.01	INSERT INTO "WORK_ON" VALUES (123456789, 1, 32.5)	1 row(s) inserted.	1
24	0.00	INSERT INTO "WORK_ON" VALUES (123456789, 2, 7.5)	1 row(s) inserted.	1
25	0.00	INSERT INTO "WORK_ON" VALUES (666884444, 3, 40.0)	1 row(s) inserted.	1
26	0.00	INSERT INTO "WORK_ON" VALUES (453453453, 1, 20.0)	1 row(s) inserted.	1
27	0.00	INSERT INTO "WORK_ON" VALUES (453453453, 2, 20.0)	1 row(s) inserted.	1
28	0.00	INSERT INTO "WORK_ON" VALUES (333445555, 2, 10.0)	1 row(s) inserted.	1
29	0.00	INSERT INTO "WORK_ON" VALUES (333445555, 3, 10.0)	1 row(s) inserted.	1
30	0.01	INSERT INTO "WORK_ON" VALUES (333445555, 10, 10.0)	1 row(s) inserted.	1
31	0.00	INSERT INTO "WORK_ON" VALUES (333445555, 20, 10.0)	1 row(s) inserted.	1

Figure 7: table1

Number	Elapsed	Statement	Feedback	Rows
32	0.00	INSERT INTO "WORK_ON" VALUES (999887777, 30, 30.0)	1 row(s) inserted.	1
33	0.00	INSERT INTO "WORK_ON" VALUES (999887777, 10, 10.0)	1 row(s) inserted.	1
34	0.00	INSERT INTO "WORK_ON" VALUES (987987987, 10, 35.0)	1 row(s) inserted.	1
35	0.00	INSERT INTO "WORK_ON" VALUES (987987987, 30, 5.0)	1 row(s) inserted.	1
36	0.00	INSERT INTO "WORK_ON" VALUES (987654321, 30, 20.0)	1 row(s) inserted.	1
37	0.01	INSERT INTO "WORK_ON" VALUES (987654321, 20, 15.0)	1 row(s) inserted.	1
38	0.00	INSERT INTO "WORK_ON" VALUES (888665555, 20, NULL)	1 row(s) inserted.	1
39	0.02	INSERT INTO PROJECT VALUES (1, 'ProductX', 'Bellaire', 5)	1 row(s) inserted.	1
40	0.00	INSERT INTO PROJECT VALUES (2, 'ProductY', 'Sugarland', 5)	1 row(s) inserted.	1
41	0.00	INSERT INTO PROJECT VALUES (3, 'ProductZ', 'Houston', 5)	1 row(s) inserted.	1
42	0.01	INSERT INTO PROJECT VALUES (10, 'Computerization', 'Sta ord	1 row(s) inserted.	1
43	0.00	INSERT INTO PROJECT VALUES (20, 'Reorganization', 'Houston',	1 row(s) inserted.	1
44	0.00	INSERT INTO PROJECT VALUES (30, 'Newbene ts', 'Sta ord', 4	1 row(s) inserted.	1
45	0.02	INSERT INTO DEPENDENT VALUES ('Alice', 333445555, 'F', '04-0	1 row(s) inserted.	1
46	0.00	INSERT INTO DEPENDENT VALUES ('Theodore', 333445555, 'M', '1	1 row(s) inserted.	1
47	0.01	INSERT INTO DEPENDENT VALUES ('Joy', 333445555, 'F', '05-03-	1 row(s) inserted.	1
48	0.00	INSERT INTO DEPENDENT VALUES ('Aoner', 987654321, 'M', '02-2	1 row(s) inserted.	1
49	0.00	INSERT INTO DEPENDENT VALUES ('Michael', 123456789, 'M', '01	1 row(s) inserted.	1
50	0.01	INSERT INTO DEPENDENT VALUES ('Alice', 123456789, 'F', '12-3	1 row(s) inserted.	1
51	0.00	INSERT INTO DEPENDENT VALUES ('Elizabeth', 123456789, 'F', '	1 row(s) inserted.	1
52	0.05	alter table employee add constraint employee_mgr foreign	Table altered.	0
53	0.02	alter table employee add constraint employee_dno foreign	Table altered.	0
54	0.02	alter table department add constraint department_mgr for	Table altered.	0
55	0.02	alter table dept_locations add constraint dept_locations	Table altered.	0
56	0.02	alter table project add constraint project_dnum foreign	Table altered.	0
57	0.02	alter table project add constraint project_plocation for	Table altered.	0
58	0.02	alter table work_on add constraint work_on_ssn foreign k	Table altered.	0
59	0.02	alter table work_on add constraint work_on_p_no foreign	Table altered.	0
60	0.01	alter table dependent add constraint dependent_ssn forei	Table altered.	0

[Download](#)

row(s) 1 - 60 of 60

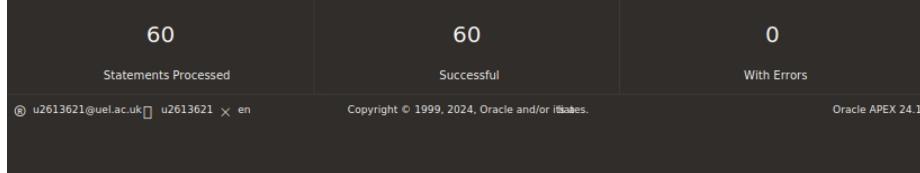
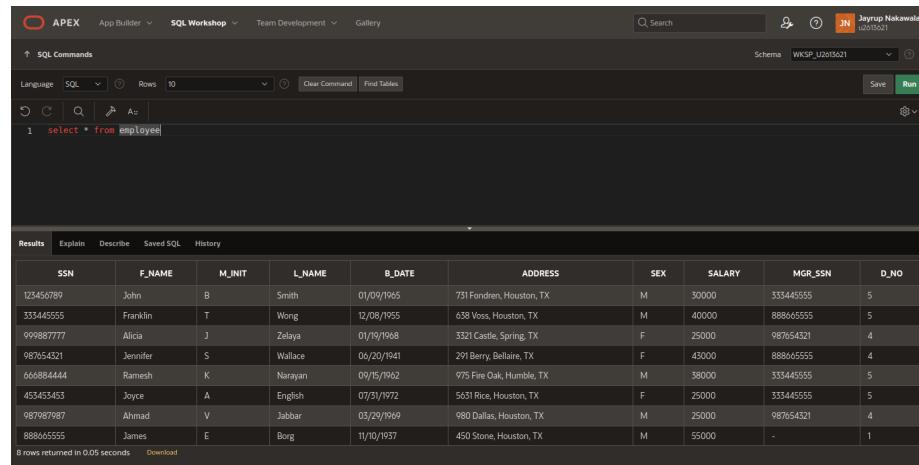


Figure 8: table2

6.2 Use basic select statements to provide rows of each table and provide screenshot.



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

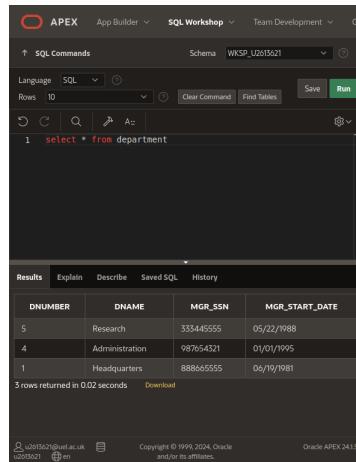
```
1 select * from employees
```

The results section displays the following data:

SSN	F_NAME	M_INIT	L_NAME	B_DATE	ADDRESS	SEX	SALARY	MGR_SSN	D_NO
123456789	John	B	Smith	01/09/1965	731 Fondren, Houston, TX	M	30000	333445555	5
333445555	Franklin	T	Wong	12/08/1955	638 Voss, Houston, TX	M	40000	888665555	5
999887777	Alicia	J	Zelaya	01/19/1968	3321 Castle, Spring, TX	F	25000	987654321	4
987654321	Jennifer	S	Wallace	06/20/1941	291 Berry, Bellaire, TX	F	43000	888665555	4
666884444	Ramesh	K	Narayan	09/15/1962	975 Fire Oak, Humble, TX	M	58000	333445555	5
454545453	Joyce	A	English	07/31/1972	5631 Rice, Houston, TX	F	25000	333445555	5
987987987	Ahmad	V	Jabbar	03/29/1969	980 Dallas, Houston, TX	M	25000	987654321	4
888665555	James	E	Borg	11/10/1937	450 Stone, Houston, TX	M	55000	-	1

8 rows returned in 0.05 seconds [Download](#)

Figure 9: table3



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

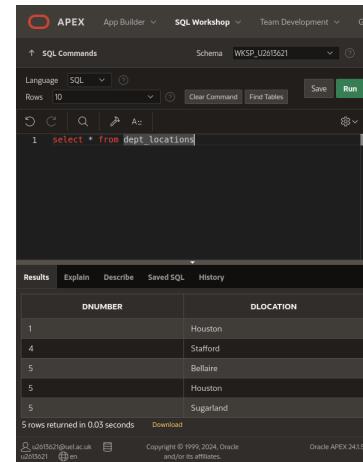
```
1 select * from department
```

The results section displays the following data:

DNUMBER	DNAME	MGR_SSN	MGR_START_DATE
5	Research	333445555	05/22/1988
4	Administration	987654321	01/01/1995
1	Headquarters	888665555	06/10/1981

3 rows returned in 0.02 seconds [Download](#)

Figure 10: Table 5



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

```
1 select * from dept_location
```

The results section displays the following data:

DNUMBER	DLOCATION
1	Houston
4	Stafford
5	Bellaire
5	Houston
5	Sugarland

5 rows returned in 0.03 seconds [Download](#)

Figure 11: Table 6

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

```
1 select * from project
```

The results table is as follows:

PNUMBER	PNAME	LOCATION	DNUM
1	ProductX	Bellaire	5
2	ProductY	Sugarland	5
5	ProductZ	Houston	5
10	Computerization	Stafford	4
20	Reorganization	Houston	1
30	Newbenefits	Stafford	4

6 rows returned in 0.01 seconds

Figure 12: Table 7

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

```
1 select * from work_on
```

The results table is as follows:

E_SSN	P_NO	HOURS_WORKED
123456789	1	32.5
123456789	2	7.5
000884444	3	40
453454545	1	20
453454545	2	20
333445555	2	10
333445555	5	10
333445555	10	10
333445555	20	10
999887777	30	30

More than 10 rows available. Increase rows selector to view more rows.

Figure 13: Table 8

The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command entered is:

```
1 select * from dependent
```

The results table is as follows:

DEP_NAME	E_SSN	SEX	B_DATE	RELATIONSHIP
Alice	333445555	F	04/05/1986	Daughter
Theodore	333445555	M	10/25/1983	Son
Joy	333445555	F	05/03/1958	Spouse
Abner	987654321	M	02/28/1942	Spouse
Michael	123456789	M	01/04/1988	Son
Alice	123456789	F	12/30/1988	Daughter
Elizabeth	123456789	F	05/05/1967	Spouse

7 rows returned in 0.03 seconds

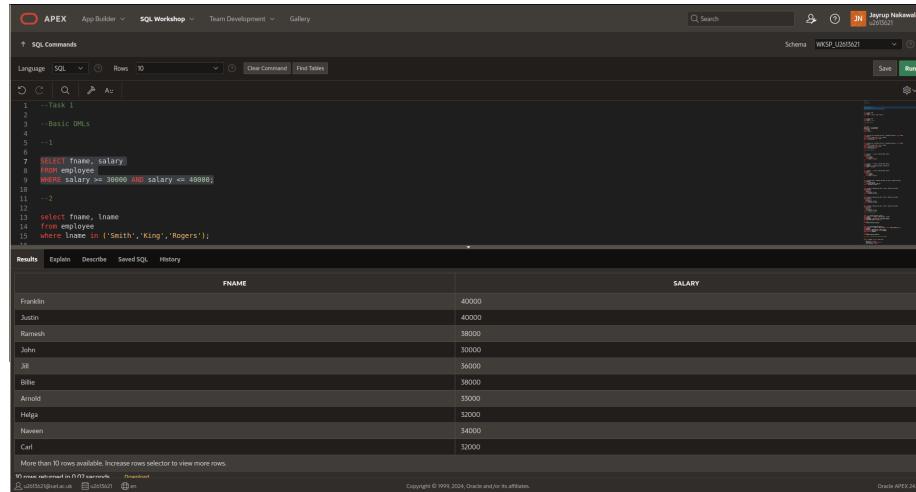
Figure 14: table9

7 Week 7

7.1 Queries

- 7.1.1 Create a SQL statement that displays only the first_name and salary of an employee whose salary is between 30,000 and 40000.

```
SELECT fname, salary
FROM employee
WHERE salary >= 30000 AND salary <= 40000;
```



The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'SQL Workshop' is selected. The schema dropdown shows 'WKSP_1285602'. The main area has 'SQL' selected in the language dropdown. The SQL editor contains the following code:

```
1 --Task 1
2
3 --Basic DMLs
4
5 --1
6
7 SELECT fname, salary
8 FROM employee
9 WHERE salary >= 30000 AND salary <= 40000;
10
11 --2
12
13 select fname, lname
14 from employee
15 where lname in ('Smith','King','Rogers');
```

The 'Results' tab is active, displaying the output of the first query:

FNAME	SALARY
Franklin	40000
Justin	40000
Ramosh	38000
John	30000
Jill	36000
Billie	38000
Arnold	35000
Hilda	32000
Navene	34000
Carl	32000

A message at the bottom of the results table says 'More than 10 rows available. Increase rows selector to view more rows.'

Figure 15: query1

- 7.1.2 Create a SQL statement that displays the first_name and last_name of employees whose last_name is either 'Smith', 'King', or 'Rogers'.

```
select fname, lname
from employee
where lname in ('Smith','King','Rogers');
```

```

1 -- Task 1
2 -- Basic DML
3 -- -1
4
5   SELECT fname, salary
6     FROM employee
7    WHERE salary >= 30000 AND salary <= 40000;
8
9
10  /*
11   * -2
12   */
13
14   SELECT fname, lname
15     FROM employee
16    WHERE lname IN ('Smith', 'King', 'Rogers');
17
18  /*
19   * -3
20   */
21
22   SELECT fname, lname
23     FROM employee
24    WHERE lname LIKE 'S%';

```

Results

FNAME	LNAME
John	Smith
Kate	King
Mike	King
Ray	King

4 rows returned in 0.01 seconds [Download](#)

Figure 16: query2

7.1.3 Create a SQL statement that displays the first_name and last_name of employees whose last_name starts with 'S'.

```

select fname, lname
from employee
where lname like 'S%';

```

```

1 -- Task 1
2 -- Basic DML
3 -- -1
4
5   SELECT fname, lname
6     FROM employee
7    WHERE lname LIKE 'S%';
8
9
10  /*
11   * -2
12   */
13
14   SELECT fname, lname
15     FROM employee
16    WHERE lname LIKE 'S%';
17
18  /*
19   * -3
20   */
21
22
23   /*
24   * -4
25   */
26
27
28   /*
29   * -5
30   */
31
32
33   /*
34   * -6
35   */
36
37
38   /*
39   * -7
40   */
41
42
43   /*
44   * -8
45   */
46
47
48   /*
49   * -9
50   */
51
52
53   /*
54   * -10
55   */
56
57
58   /*
59   * -11
60   */
61
62
63   /*
64   * -12
65   */
66
67
68   /*
69   * -13
70   */
71
72
73   /*
74   * -14
75   */
76
77
78   /*
79   * -15
80   */
81
82
83   /*
84   * -16
85   */
86
87
88   /*
89   * -17
90   */
91
92
93   /*
94   * -18
95   */
96
97
98   /*
99   * -19
100  */
101
102
103   /*
104  * -20
105  */
106
107
108   /*
109  * -21
110  */
111
112
113   /*
114  * -22
115  */
116
117
118   /*
119  * -23
120  */
121
122
123   /*
124  * -24
125  */
126
127
128   /*
129  * -25
130  */
131
132
133   /*
134  * -26
135  */
136
137
138   /*
139  * -27
140  */
141
142
143   /*
144  * -28
145  */
146
147
148   /*
149  * -29
150  */
151
152
153   /*
154  * -30
155  */
156
157
158   /*
159  * -31
160  */
161
162
163   /*
164  * -32
165  */
166
167
168   /*
169  * -33
170  */
171
172
173   /*
174  * -34
175  */
176
177
178   /*
179  * -35
180  */
181
182
183   /*
184  * -36
185  */
186
187
188   /*
189  * -37
190  */
191
192
193   /*
194  * -38
195  */
196
197
198   /*
199  * -39
200  */
201
202
203   /*
204  * -40
205  */
206
207
208   /*
209  * -41
210  */
211
212
213   /*
214  * -42
215  */
216
217
218   /*
219  * -43
220  */
221
222
223   /*
224  * -44
225  */
226
227
228   /*
229  * -45
230  */
231
232
233   /*
234  * -46
235  */
236
237
238   /*
239  * -47
240  */
241
242
243   /*
244  * -48
245  */
246
247
248   /*
249  * -49
250  */
251
252
253   /*
254  * -50
255  */
256
257
258   /*
259  * -51
260  */
261
262
263   /*
264  * -52
265  */
266
267
268   /*
269  * -53
270  */
271
272
273   /*
274  * -54
275  */
276
277
278   /*
279  * -55
280  */
281
282
283   /*
284  * -56
285  */
286
287
288   /*
289  * -57
290  */
291
292
293   /*
294  * -58
295  */
296
297
298   /*
299  * -59
300  */
301
302
303   /*
304  * -60
305  */
306
307
308   /*
309  * -61
310  */
311
312
313   /*
314  * -62
315  */
316
317
318   /*
319  * -63
320  */
321
322
323   /*
324  * -64
325  */
326
327
328   /*
329  * -65
330  */
331
332
333   /*
334  * -66
335  */
336
337
338   /*
339  * -67
340  */
341
342
343   /*
344  * -68
345  */
346
347
348   /*
349  * -69
350  */
351
352
353   /*
354  * -70
355  */
356
357
358   /*
359  * -71
360  */
361
362
363   /*
364  * -72
365  */
366
367
368   /*
369  * -73
370  */
371
372
373   /*
374  * -74
375  */
376
377
378   /*
379  * -75
380  */
381
382
383   /*
384  * -76
385  */
386
387
388   /*
389  * -77
390  */
391
392
393   /*
394  * -78
395  */
396
397
398   /*
399  * -79
400  */
401
402
403   /*
404  * -80
405  */
406
407
408   /*
409  * -81
410  */
411
412
413   /*
414  * -82
415  */
416
417
418   /*
419  * -83
420  */
421
422
423   /*
424  * -84
425  */
426
427
428   /*
429  * -85
430  */
431
432
433   /*
434  * -86
435  */
436
437
438   /*
439  * -87
440  */
441
442
443   /*
444  * -88
445  */
446
447
448   /*
449  * -89
450  */
451
452
453   /*
454  * -90
455  */
456
457
458   /*
459  * -91
460  */
461
462
463   /*
464  * -92
465  */
466
467
468   /*
469  * -93
470  */
471
472
473   /*
474  * -94
475  */
476
477
478   /*
479  * -95
480  */
481
482
483   /*
484  * -96
485  */
486
487
488   /*
489  * -97
490  */
491
492
493   /*
494  * -98
495  */
496
497
498   /*
499  * -99
500  */
501
502
503   /*
504  * -100
505  */
506
507
508   /*
509  * -101
510  */
511
512
513   /*
514  * -102
515  */
516
517
518   /*
519  * -103
520  */
521
522
523   /*
524  * -104
525  */
526
527
528   /*
529  * -105
530  */
531
532
533   /*
534  * -106
535  */
536
537
538   /*
539  * -107
540  */
541
542
543   /*
544  * -108
545  */
546
547
548   /*
549  * -109
550  */
551
552
553   /*
554  * -110
555  */
556
557
558   /*
559  * -111
560  */
561
562
563   /*
564  * -112
565  */
566
567
568   /*
569  * -113
570  */
571
572
573   /*
574  * -114
575  */
576
577
578   /*
579  * -115
580  */
581
582
583   /*
584  * -116
585  */
586
587
588   /*
589  * -117
590  */
591
592
593   /*
594  * -118
595  */
596
597
598   /*
599  * -119
600  */
601
602
603   /*
604  * -120
605  */
606
607
608   /*
609  * -121
610  */
611
612
613   /*
614  * -122
615  */
616
617
618   /*
619  * -123
620  */
621
622
623   /*
624  * -124
625  */
626
627
628   /*
629  * -125
630  */
631
632
633   /*
634  * -126
635  */
636
637
638   /*
639  * -127
640  */
641
642
643   /*
644  * -128
645  */
646
647
648   /*
649  * -129
650  */
651
652
653   /*
654  * -130
655  */
656
657
658   /*
659  * -131
660  */
661
662
663   /*
664  * -132
665  */
666
667
668   /*
669  * -133
670  */
671
672
673   /*
674  * -134
675  */
676
677
678   /*
679  * -135
680  */
681
682
683   /*
684  * -136
685  */
686
687
688   /*
689  * -137
690  */
691
692
693   /*
694  * -138
695  */
696
697
698   /*
699  * -139
700  */
701
702
703   /*
704  * -140
705  */
706
707
708   /*
709  * -141
710  */
711
712
713   /*
714  * -142
715  */
716
717
718   /*
719  * -143
720  */
721
722
723   /*
724  * -144
725  */
726
727
728   /*
729  * -145
730  */
731
732
733   /*
734  * -146
735  */
736
737
738   /*
739  * -147
740  */
741
742
743   /*
744  * -148
745  */
746
747
748   /*
749  * -149
750  */
751
752
753   /*
754  * -150
755  */
756
757
758   /*
759  * -151
760  */
761
762
763   /*
764  * -152
765  */
766
767
768   /*
769  * -153
770  */
771
772
773   /*
774  * -154
775  */
776
777
778   /*
779  * -155
780  */
781
782
783   /*
784  * -156
785  */
786
787
788   /*
789  * -157
790  */
791
792
793   /*
794  * -158
795  */
796
797
798   /*
799  * -159
800  */
801
802
803   /*
804  * -160
805  */
806
807
808   /*
809  * -161
810  */
811
812
813   /*
814  * -162
815  */
816
817
818   /*
819  * -163
820  */
821
822
823   /*
824  * -164
825  */
826
827
828   /*
829  * -165
830  */
831
832
833   /*
834  * -166
835  */
836
837
838   /*
839  * -167
840  */
841
842
843   /*
844  * -168
845  */
846
847
848   /*
849  * -169
850  */
851
852
853   /*
854  * -170
855  */
856
857
858   /*
859  * -171
860  */
861
862
863   /*
864  * -172
865  */
866
867
868   /*
869  * -173
870  */
871
872
873   /*
874  * -174
875  */
876
877
878   /*
879  * -175
880  */
881
882
883   /*
884  * -176
885  */
886
887
888   /*
889  * -177
890  */
891
892
893   /*
894  * -178
895  */
896
897
898   /*
899  * -179
900  */
901
902
903   /*
904  * -180
905  */
906
907
908   /*
909  * -181
910  */
911
912
913   /*
914  * -182
915  */
916
917
918   /*
919  * -183
920  */
921
922
923   /*
924  * -184
925  */
926
927
928   /*
929  * -185
930  */
931
932
933   /*
934  * -186
935  */
936
937
938   /*
939  * -187
940  */
941
942
943   /*
944  * -188
945  */
946
947
948   /*
949  * -189
950  */
951
952
953   /*
954  * -190
955  */
956
957
958   /*
959  * -191
960  */
961
962
963   /*
964  * -192
965  */
966
967
968   /*
969  * -193
970  */
971
972
973   /*
974  * -194
975  */
976
977
978   /*
979  * -195
980  */
981
982
983   /*
984  * -196
985  */
986
987
988   /*
989  * -197
990  */
991
992
993   /*
994  * -198
995  */
996
997
998   /*
999  * -199
1000 */
1001
1002
1003   /*
1004  * -200
1005  */
1006
1007
1008   /*
1009  * -201
1010  */
1011
1012
1013   /*
1014  * -202
1015  */
1016
1017
1018   /*
1019  * -203
1020  */
1021
1022
1023   /*
1024  * -204
1025  */
1026
1027
1028   /*
1029  * -205
1030  */
1031
1032
1033   /*
1034  * -206
1035  */
1036
1037
1038   /*
1039  * -207
1040  */
1041
1042
1043   /*
1044  * -208
1045  */
1046
1047
1048   /*
1049  * -209
1050  */
1051
1052
1053   /*
1054  * -210
1055  */
1056
1057
1058   /*
1059  * -211
1060  */
1061
1062
1063   /*
1064  * -212
1065  */
1066
1067
1068   /*
1069  * -213
1070  */
1071
1072
1073   /*
1074  * -214
1075  */
1076
1077
1078   /*
1079  * -215
1080  */
1081
1082
1083   /*
1084  * -216
1085  */
1086
1087
1088   /*
1089  * -217
1090  */
1091
1092
1093   /*
1094  * -218
1095  */
1096
1097
1098   /*
1099  * -219
1100  */
1101
1102
1103   /*
1104  * -220
1105  */
1106
1107
1108   /*
1109  * -221
1110  */
1111
1112
1113   /*
1114  * -222
1115  */
1116
1117
1118   /*
1119  * -223
1120  */
1121
1122
1123   /*
1124  * -224
1125  */
1126
1127
1128   /*
1129  * -225
1130  */
1131
1132
1133   /*
1134  * -226
1135  */
1136
1137
1138   /*
1139  * -227
1140  */
1141
1142
1143   /*
1144  * -228
1145  */
1146
1147
1148   /*
1149  * -229
1150  */
1151
1152
1153   /*
1154  * -230
1155  */
1156
1157
1158   /*
1159  * -231
1160  */
1161
1162
1163   /*
1164  * -232
1165  */
1166
1167
1168   /*
1169  * -233
1170  */
1171
1172
1173   /*
1174  * -234
1175  */
1176
1177
1178   /*
1179  * -235
1180  */
1181
1182
1183   /*
1184  * -236
1185  */
1186
1187
1188   /*
1189  * -237
1190  */
1191
1192
1193   /*
1194  * -238
1195  */
1196
1197
1198   /*
1199  * -239
1200  */
1201
1202
1203   /*
1204  * -240
1205  */
1206
1207
1208   /*
1209  * -241
1210  */
1211
1212
1213   /*
1214  * -242
1215  */
1216
1217
1218   /*
1219  * -243
1220  */
1221
1222
1223   /*
1224  * -244
1225  */
1226
1227
1228   /*
1229  * -245
1230  */
1231
1232
1233   /*
1234  * -246
1235  */
1236
1237
1238   /*
1239  * -247
1240  */
1241
1242
1243   /*
1244  * -248
1245  */
1246
1247
1248   /*
1249  * -249
1250  */
1251
1252
1253   /*
1254  * -250
1255  */
1256
1257
1258   /*
1259  * -251
1260  */
1261
1262
1263   /*
1264  * -252
1265  */
1266
1267
1268   /*
1269  * -253
1270  */
1271
1272
1273   /*
1274  * -254
1275  */
1276
1277
1278   /*
1279  * -255
1280  */
1281
1282
1283   /*
1284  * -256
1285  */
1286
1287
1288   /*
1289  * -257
1290  */
1291
1292
1293   /*
1294  * -258
1295  */
1296
1297
1298   /*
1299  * -259
1300  */
1301
1302
1303   /*
1304  * -260
1305  */
1306
1307
1308   /*
1309  * -261
1310  */
1311
1312
1313   /*
1314  * -262
1315  */
1316
1317
1318   /*
1319  * -263
1320  */
1321
1322
1323   /*
1324  * -264
1325  */
1326
1327
1328   /*
1329  * -265
1330  */
1331
1332
1333   /*
1334  * -266
1335  */
1336
1337
1338   /*
1339  * -267
1340  */
1341
1342
1343   /*
1344  * -268
1345  */
1346
1347
1348   /*
1349  * -269
1350  */
1351
1352
1353   /*
1354  * -270
1355  */
1356
1357
1358   /*
1359  * -271
1360  */
1361
1362
1363   /*
1364  * -272
1365  */
1366
1367
1368   /*
1369  * -273
1370  */
1371
1372
1373   /*
1374  * -274
1375  */
1376
1377
1378   /*
1379  * -275
1380  */
1381
1382
1383   /*
1384  * -276
1385  */
1386
1387
1388   /*
1389  * -277
1390  */
1391
1392
1393   /*
1394  * -278
1395  */
1396
1397
1398   /*
1399  * -279
1400  */
1401
1402
1403   /*
1404  * -280
1405  */
1406
1407
1408   /*
1409  * -281
1410  */
1411
1412
1413   /*
1414  * -282
1415  */
1416
1417
1418   /*
1419  * -283
1420  */
1421
1422
1423   /*
1424  * -284
1425  */
1426
1427
1428   /*
1429  * -285
1430  */
1431
1432
1433   /*
1434  * -286
1435  */
1436
1437
1438   /*
1439  * -287
1440  */
1441
1442
1443   /*
1444  * -288
1445  */
1446
1447
1448   /*
1449  * -289
1450  */
1451
1452
1453   /*
1454  * -290
1455  */
1456
1457
1458   /*
1459  * -291
1460  */
1461
1462
1463   /*
1464  * -292
1465  */
1466
1467
1468   /*
1469  * -293
1470  */
1471
1472
1473   /*
1474  * -294
1475  */
1476
1477
1478   /*
1479  * -295
1480  */
1481
1482
1483   /*
1484  * -296
1485  */
1486
1487
1488   /*
1489  * -297
1490  */
1491
1492
1493   /*
1494  * -298
1495  */
1496
1497
1498   /*
1499  * -299
1500  */
1501
1502
1503   /*
1504  * -300
1505  */
1506
1507
1508   /*
1509  * -301
1510  */
1511
1512
1513   /*
1514  * -302
1515  */
1516
1517
1518   /*
1519  * -303
1520  */
1521
1522
1523   /*
1524  * -304
1525  */
1526
1527
1528   /*
1529  * -305
1530  */
1531
1532
1533   /*
1534  * -306
1535  */
1536
1537
1538   /*
1539  * -307
1540  */
1541
1542
1543   /*
1544  * -308
1545  */
1546
1547
1548   /*
1549  * -309
1550  */
1551
1552
1553   /*
1554  * -310
1555  */
1556
1557
1558   /*
1559  * -311
1560  */
1561
1562
1563   /*
1564  * -312
1565  */
1566
1567
1568   /*
1569  * -313
1570  */
1571
1572
1573   /*
1574  * -314
1575  */
1576
1577
1578   /*
1579  * -315
1580  */
1581
1582
1583   /*
1584  * -316
1585  */
1586
1587
1588   /*
1589  * -317
1590  */
1591
1592
1593   /*
1594  * -318
1595  */
1596
1597
1598   /*
1599  * -319
1600  */
1601
1602
1603   /*
1604  * -320
1605  */
1606
1607
1608   /*
1609  * -321
1610  */
1611
1612
1613   /*
1614  * -322
1615  */
1616
1617
1618   /*
1619  * -323
1620  */
1621
1622
1623   /*
1624  * -324
1625  */
1626
1627
1628   /*
1629  * -325
1630  */
1631
1632
1633   /*
1634  * -326
1635  */
1636
1637
1638   /*
1639  * -327
1640  */
1641
1642
1643   /*
1644  * -328
1645  */
1646
1647
1648   /*
1649  * -329
1650  */
1651
1652
1653   /*
1654  * -330
1655  */
1656
1657
1658   /*
1659  * -331
1660  */
1661
1662
1663   /*
1664  * -332
1665  */
1666
1667
1668   /*
1669  * -333
1670  */
1671
1672
1673   /*
1674  * -334
1675  */
1676
1677
1678   /*
1679  * -335
1680  */
1681
1682
1683   /*
1684  * -336
1685  */
1686
1687
1688   /*
1689  * -337
1690  */
1691
1692
1693   /*
1694  * -338
1695  */
1696
1697
1698   /*
1699  * -339
1700  */
1701
1702
1703   /*
1704  * -340
1705  */
1706
1707
1708   /*
1709  * -341
1710  */
1711
1712
1713   /*
1714  * -342
1715  */
1716
1717
1718   /*
1719  * -343
1720  */
1721
1722
1723   /*
1724  * -344
1725  */
1726
1727
1728   /*
1729  * -345
1730  */
1731
1732
1733   /*
1734  * -346
1735  */
1736
1737
1738   /*
1739  * -347
1740  */
1741
1742
1743   /*
1744  * -348
1745  */
1746
1747
1748   /*
1749  * -349
1750  */
1751
1752
1753   /*
1754  * -350
1755  */
1756
1757
1758   /*
1759  * -351
1760  */
1761
1762
1763   /*
1764  * -352
1765  */
1766
1767
1768   /*
1769  * -353
1770  */
1771
1772
1773   /*
1774  * -354
1775  */
1776
1777
1778   /*
1779  * -355
1780  */
1781
1782
1783   /*
1784  * -356
1785  */
1786
1787
1788   /*
1789  * -357
1790  */
1791
1792
1793   /*
1794  * -358
1795  */
1796
1797
1798   /*
1799  * -359
1800  */
1801
1802
1803   /*
1804  * -360
1805  */
1806
1807
1808   /*
1809  * -361
1810  */
1811
1812
1813   /*
1814  * -362
1815  */
1816
1817
1818   /*
1819
```

7.1.4 For each department, retrieve the department number, the number of employees in the department, and their average salary.

```
select dno,
       count(ssn) as no_of_employees,
       avg(salary) as average_salary
  from employee
 group by dno
 order by dno asc;
```

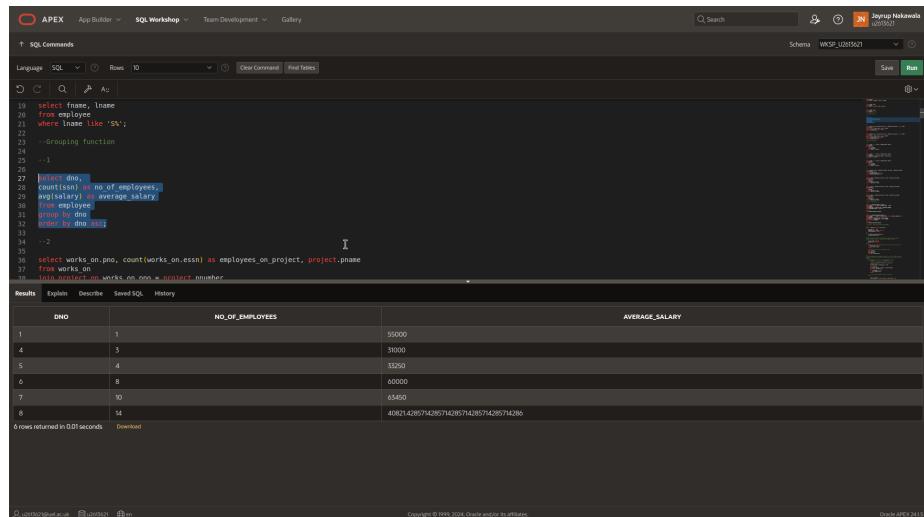


Figure 18: query4

7.1.5 For each project, retrieve the project number, the project name, and the number of employees who work on that project.

```
select works_on.pno, count(works_on.essn) as employees_on_project, project.pname
from works_on
join project on works_on.pno = project.pnumber
GROUP BY works_on.pno, project.pname
Order by works_on.pno asc;
```

PNO	EMPLOYEES_ON_PROJECT	PNAME
1	2	ProductX
2	5	ProductY
3	2	ProductZ
10	5	Computerization
20	5	Reorganization
30	5	NewBenefits
61	9	OperatingSystems
62	8	DatabaseSystems
63	4	Middleware
91	8	PrinterManufacturing

Figure 19: query5

- 7.1.6 For each project on which more than two employees work, retrieve the project number, the project name, and the number of employees who work on the project.**

```

select works_on.pno, count(works_on.essn) as employees_on_project, project.pname
from works_on
join project on works_on.pno = project.pnumber
GROUP BY works_on.pno, project.pname
having count(works_on.essn) > 2
Order by works_on.pno asc;

```

PNO	EMPLOYEES_ON_PROJECT	PHNAME
2	5	ProductY
10	5	Computerization
20	5	Reorganization
50	5	NewBenefits
61	0	OperatingSystems
62	8	DatabaseSystems
65	4	Middleware
91	8	ImagePrinters
92	5	LaserPrinters

Figure 20: query6

7.1.7 Retrieve the name and address of all employees who work for the ‘Research’ department without using join

```
select fname ||' '| lname as Employee_Name, address
from employee
where dno in (
    select dnumber
    from department
    where dname = 'Research'
);
```

EMPLOYEE_NAME	ADDRESS
Franklin Wong	658 Voss, Houston, TX
Joyce English	5631 Rice, Houston, TX
Ramanesh Narayanan	971 Fire Oak, Humble, TX
John Smith	731 Fondren, Houston, TX

Figure 21: query7

7.1.8 Modify query 1 and observe results by using join

```
select fname ||' '| lname as Employee_Name, address  
from employee  
join department on department.dnumber = employee.dno  
where dname = 'Research'
```

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Commands' tab is selected. The SQL editor contains the modified query:

```
50 | select fname ||' '| lname as Employee_Name, address  
51 | from employee  
52 | join department on department.dnumber = employee.dno  
53 | where dname = 'Research'  
54 | --2  
55 |  
56 | select fname ||' '| lname as Employee_Name, address  
57 | from employee  
58 | join department on department.dnumber = employee.dno  
59 | where dname = 'Research'  
60 | --3  
61 |  
62 | select fname ||' '| lname as Employee_Name, address  
63 | from employee  
64 | join department on department.dnumber = employee.dno  
65 | where dname = 'Research'  
66 |  
67 |  
68 |  
69 |  
70 |  
71 |  
72 |  
73 |  
74 |  
75 |  
76 |  
77 |  
78 |
```

The 'Results' tab is selected, displaying the query output:

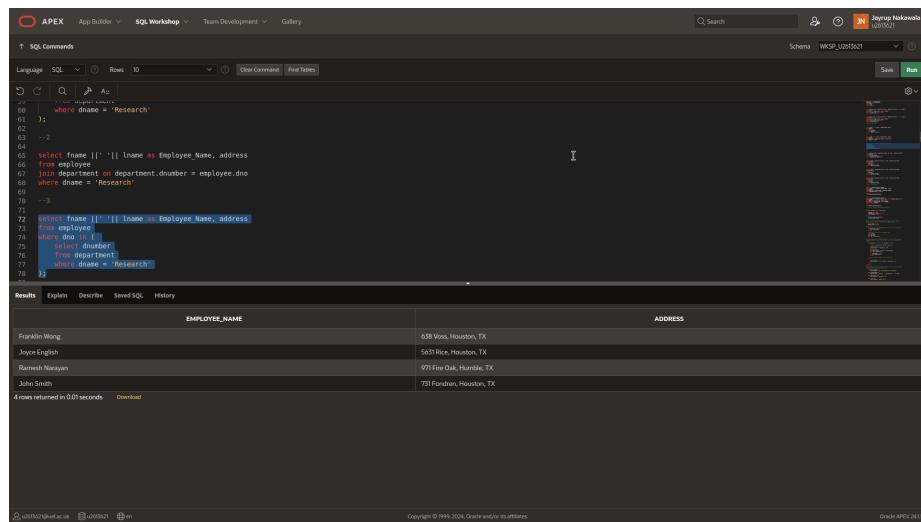
EMPLOYEE_NAME	ADDRESS
Franklin Wong	588 Voss, Houston, TX
Joyce English	501 Rice, Houston, TX
Karnish Narayan	971 Elm Oak, Humble, TX
John Smith	731 Fondren, Houston, TX

Below the table, it says '4 rows returned in 0.01 seconds'.

Figure 22: query8

7.1.9 Modify query 1 and use subquery instead of join

```
select fname ||' '| lname as Employee_Name, address
from employee
where dno in (
    select dnumber
    from department
    where dname = 'Research'
);
```



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The user 'Jayap Nukarla' is logged in. The schema is set to 'WKSP_JNBST&CT'. The SQL Commands tab is active, showing the modified query:

```
select fname ||' '| lname as Employee_Name, address
from employee
where dno in (
    select dnumber
    from department
    where dname = 'Research'
);
```

The results tab displays the output of the query:

EMPLOYEE_NAME	ADDRESS
Franklin Wong	638 Voss, Houston, TX
Joyce English	5671 Rice, Houston, TX
Karnish Narayan	9713ine Oak, Humble, TX
John Smith	731 Fondren, Houston, TX

4 rows returned in 0.01 seconds. There are links for 'Download' and 'Copy'.

Figure 23: query9

7.1.10 Retrieve the last name of employees and their supervisors (using inner join)

```
select employee.lname as Employee_Last_Name, mgr.lname as Manager_Last_Name
from employee
inner join employee mgr
    on employee.superssn = mgr.ssn
order by Employee_Last_Name asc;
```

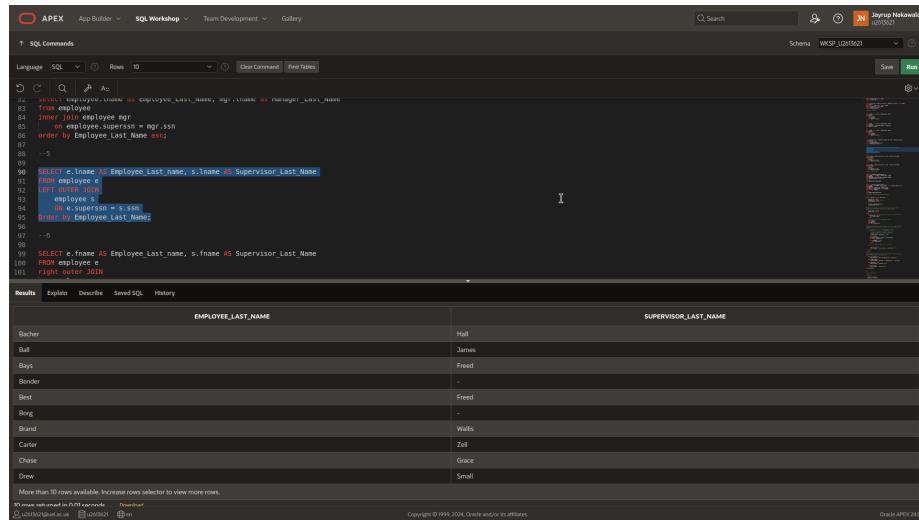
The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command window contains the query from the previous step. The results window displays two columns: EMPLOYEE_LAST_NAME and MANAGER_LAST_NAME. The data is as follows:

EMPLOYEE_LAST_NAME	MANAGER_LAST_NAME
Sachar	Hall
Ball	James
Bay	Freed
Bent	Freed
Brand	Walls
Carter	Zell
Chase	Grace
Dew	Small
English	Wong
Hall	Reedy

Figure 24: query10

7.1.11 Modify query 4 to display all employees with supervisor and also that employee where a supervisor is not assigned (left outer join) Note the difference in results by increasing row counts in the output

```
SELECT e.lname AS Employee_Last_name, s.lname AS Supervisor_Last_Name
FROM employee e
LEFT OUTER JOIN
    employee s
    ON e.superssn = s.ssn
Order by Employee_Last_Name;
```



The screenshot shows the Oracle SQL Workshop interface. The SQL command window contains the query provided above. The results window displays two columns: 'EMPLOYEE_LAST_NAME' and 'SUPERVISOR_LAST_NAME'. The data is as follows:

EMPLOYEE_LAST_NAME	SUPERVISOR_LAST_NAME
Bachar	Hall
Ball	James
Bayes	Freed
Bender	-
Bert	Freed
Big	-
Blund	Walls
Carter	Zoll
Chase	Grace
Drew	Small

A note at the bottom of the results table says: "More than 10 rows available. Increase rows selector to view more rows."

Figure 25: query11

7.1.12 Modify query 4 to display all the supervisors with and without employees assigned to them (Right outer join)

```

SELECT e.fname AS Employee_Last_name, s.fname AS Supervisor_Last_Name
FROM employee e
right outer JOIN
    employee s
    ON e.superssn = s.ssn
Order by Employee_Last_Name;

```

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. The schema is set to WMSPLDST&CT. The SQL Commands tab is active, displaying the following code:

```

88 --5
89
90     SELECT e.lname AS Employee_Last_name, s.lname AS Supervisor_Last_Name
91   FROM employee e
92  LEFT OUTER JOIN
93      employee s
94  ON e.superssn = s.ssn
95  ORDER BY Employee_Last_Name;
96
97 --6
98
99     SELECT e.fname AS Employee_Last_name, s.fname AS Supervisor_Last_Name
100  FROM employee e
101  RIGHT OUTER JOIN
102      employee s
103  ON e.superssn = s.ssn
104  ORDER BY Employee_Last_Name;
105
106 --7
107

```

The results tab displays the output of the query:

EMPLOYEE_LAST_NAME	SUPERVISOR_LAST_NAME
Ahmed	Jennifer
Alic	Alex
Alica	Jennifer
Andy	Evan
Arnold	Kate
Steve	Lyle
Bonnie	Alex
Brad	Jared
Carl	Nozzen
Chris	John

More than 10 rows available. Increase rows selector to view more rows.

Figure 26: query12

7.1.13 Modify query 4 to display all employees with and without supervisors and all supervisors with and without employees (Full outer join)

```

SELECT e.fname AS Employee_Last_name, s.fname AS Supervisor_Last_Name
FROM employee e
full outer JOIN
    employee s
    ON e.superssn = s.ssn
Order by Employee_Last_Name;

```

The screenshot shows the Oracle SQL Workshop interface. The SQL Commands tab contains the following code:

```

99  SELECT e.fname AS Employee_Last_name, s.fname AS Supervisor_Last_Name
100 FROM employee e
101 full outer JOIN
102     employee s
103     ON e.superssn = s.ssn
104 Order by Employee_Last_Name;
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137

```

The Results tab displays the output of the query:

EMPLOYEE_LAST_NAME	SUPERVISOR_LAST_NAME
Ahmad	Jennifer
Alic	Alex
Alix	-
Alica	Jennifer
Andy	Evan
Arnold	Kate
Bille	Lyle
Bob	-
Bonnie	Alex
Brad	Jared

More than 10 rows available. Increase rows selector to view more rows.

Copyright © 1999-2004, Oracle and/or its affiliates.

Figure 27: query13

7.2 Creating Views

7.2.1 Create a view to display employee name and their salaries who work for the research department.

```
CREATE VIEW RESEARCH_EMPLOYEE_SALARIES AS
SELECT FNAME ||' '|| LNAME AS Employee_Name , SALARY
FROM EMPLOYEE
JOIN DEPARTMENT ON EMPLOYEE.DNO = DEPARTMENT.DNUMBER
WHERE DEPARTMENT.DNAME = 'Research';

select *
from RESEARCH_EMPLOYEE_SALARIES;
```

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'Schema' dropdown is set to 'WKSP_U015621'. The main area has tabs for 'SQL Commands' (selected), 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The SQL Commands tab contains the code for creating the 'RESEARCH_EMPLOYEE_SALARIES' view. The Results tab displays the output of the query, showing four rows of employee names and salaries:

Employee_Name	Salary
Franklin Wong	40000
Joyce English	25000
Karnesh Nayyan	30000
John Smith	30000

Below the table, it says '4 rows returned in 0.03 seconds'.

Figure 28: view 1

7.2.2 Create a view to display the employee name and the name of the project and project hour in which each employee works. Provide a screenshot of the code and results

```
CREATE VIEW RESEARCH_EMPLOYEE_PROJECTS AS
SELECT FNAME ||' '|| LNAME AS Employee_Name , PROJECT.PNAME , WORKS_ON.HOURS
FROM EMPLOYEE
JOIN WORKS_ON ON EMPLOYEE.SSN = WORKS_ON.ESSN
JOIN PROJECT ON WORKS_ON.PNO = PROJECT.PNUMBER
ORDER BY PROJECT.PNUMBER;

select *
```

```
from RESEARCH_EMPLOYEE_PROJECTS;
```

APEX App Builder SQL Workshop Team Development Gallery

Search Schema WKSP_U2B13621

SQL Commands

Language SQL Rows 10 Close Command Find Tables

```
1 JOIN DEPARTMENT ON EMPLOYEE.DNO = DEPARTMENT.DNUMBER  
2 WHERE DEPARTMENT.DNAME = 'Research';  
3  
4 select *  
5 from RESEARCH_EMPLOYEE_SALARIES;  
6 --  
7  
8  
9 CREATE VIEW RESEARCH_EMPLOYEE_PROJECTS AS  
10 SELECT PNAME || ' | ' || LNAME as Employee_Name , PNAME,PNAME, WORKS.ON.HOURS  
11 FROM PROJECT ON EMPLOYEE.SSN = WORKS.ON.ESSN  
12 JOIN WORKS ON (EMPLOYEE.SSN = WORKS.ON.ESSN  
13 JOIN PROJECT ON WORKS.ON.PNO = PROJECT.PNUMBER  
14 WHERE PNAME IN ('Computerization')  
15  
16 select *  
17 from RESEARCH_EMPLOYEE_PROJECTS;  
18  
19 --LAB TASK 3: CREATING AND EXECUTING PROCEDURES.  
20
```

Results Explain Describe Saved SQL History

EMPLOYEE_NAME	PNAME	HOURS
John Smith	ProductX	32.5
Joyce English	ProductX	20
Franklin Wong	ProductY	10
John Smith	ProductY	75
Joyce English	ProductY	20
Ramesh Narayan	ProductZ	40
Franklin Wong	ProductZ	10
Alicia Zaliza	Computerization	10
Ahmad Jabbar	Computerization	35
Franklin Wong	Computerization	10

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.000 seconds. Download

U2B13621@localhost:1521 (20250) Copyright © 1995-2018, Oracle and/or its affiliates. Oracle APEX 21

Figure 29: view 2

7.3 Creating and Executing Procedures

7.3.1 Download 1a - Raise SalaryProcedure, and run in APEX. Understand the procedure logic

```
CREATE OR REPLACE PROCEDURE raise_salary
(
    employee_ssn IN CHAR,
    employee_pct IN NUMBER DEFAULT 5,
    result_message OUT CHAR
)
AS
    old_salary employee.salary%TYPE;
    increase_amount NUMBER;

/*
Program-defined exceptions are declared here and are used to identify
exception events which will interrupt main program execution.
*/
    pct_too_high EXCEPTION;
    update_error EXCEPTION;

BEGIN
    --Disallow raises which exceed 50% on the basis of the business rules.
```

```

IF employee_pct > 50 THEN
    RAISE pct_too_high;
END IF;

--Retrieve the salary from the employee table
SELECT salary
INTO old_salary
FROM employee
WHERE ssn = employee_ssn;

IF (old_salary IS NOT NULL) AND (old_salary > 0) THEN

    increase_amount := employee_pct / 100;

    UPDATE employee
    SET salary = salary + (salary * increase_amount)
    WHERE ssn = employee_ssn;

    IF SQL%ROWCOUNT <> 1 THEN
        RAISE update_error;
    END IF;

    --Set the output parameter value if necessary.
ELSE

    --Set the message
    result_message := 'Current salary is either NULL or 0';

    END IF;

EXCEPTION
/*
Set the output parameter value here as well based upon program-defined and
system-defined exceptions which might occur.
*/
WHEN pct_too_high THEN
    result_message := 'Raise percentage may not exceed 50%';

WHEN NO_DATA_FOUND THEN
    result_message := 'Employee ' || employee_ssn || ' not found';

WHEN update_error THEN
    result_message := 'Database error';

WHEN OTHERS THEN

```

```
    result_message := 'Unknown error';  
  
END raise_salary;
```

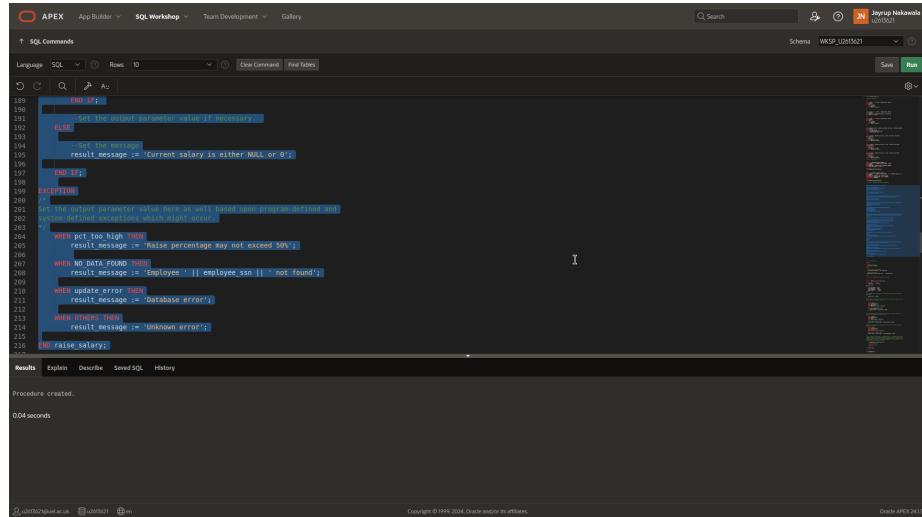


Figure 30: procedure 1

7.3.2 Execute the procedure using 1b- Fire Raise Salary procedure, to test output. Provide different values in the test procedure to test each statement in the program logic. For each test statement provide a screenshot of the code and execution results

```
DECLARE
    --Declare variables
    output_text CHAR(100);

BEGIN
    --Fire raise_salary procedure
    raise_salary('123456789', 48, output_text);

    --Output the results
    dbms_output.put_line('Output Text: ' || output_text );

END;
```

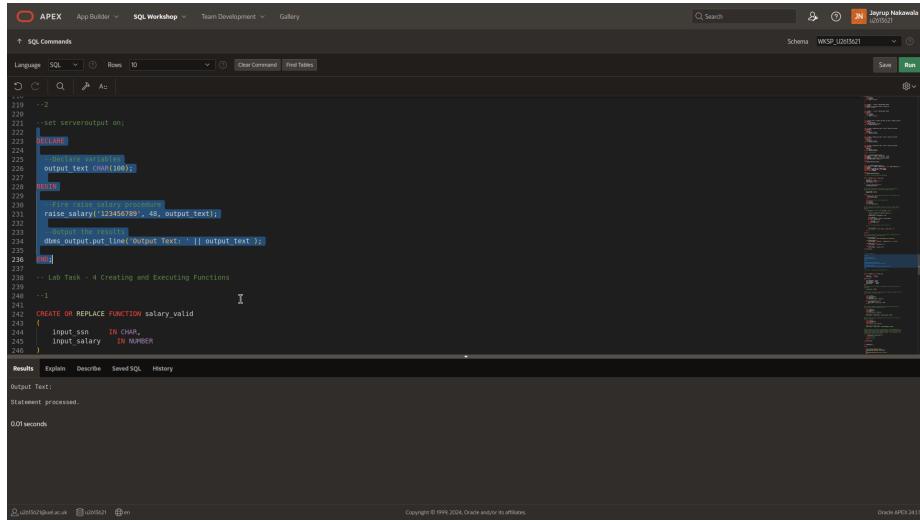


Figure 31: procedure 2

7.4 Creating and Executing Functions

7.4.1 Download 2a - Salary Valid function, and run in APEX. Understand the function logic

```
CREATE OR REPLACE FUNCTION salary_valid
(
    input_ssn      IN CHAR,
    input_salary   IN NUMBER
)
RETURN boolean
IS
    count_management NUMBER;
    count_projects   NUMBER;
    count_dependents NUMBER;
    salary_limit     NUMBER;
BEGIN
    salary_limit := 50000;
    SELECT count(*)
    INTO count_management
    FROM department
    WHERE department.mgrssn = input_ssn;
```

```

--Test the count_management value
IF count_management > 0 THEN
    salary_limit := salary_limit + 1000;
END IF;

SELECT count(*)
INTO count_projects
FROM works_on
WHERE works_on.essn = input_ssn;

--Recalculate the salary limit
salary_limit := salary_limit + (count_projects * 2000);

SELECT count(*)
INTO count_dependents
FROM dependent
WHERE dependent.essn = input_ssn;

--Recalculate the salary limit
salary_limit := salary_limit + (count_dependents * 3000);

IF input_salary > salary_limit THEN
    RETURN (FALSE);
ELSE
    RETURN (TRUE);
END IF;

END salary_valid;

```

```

APEX App Builder SQL Workshop Team Development Gallery
SQL Commands Language: SQL Rows: 10 Clear Command Find Table Save Run
789 1000 per_dependent
790 1001
791 1002 SELECT count(*)
792 1003 INTO count_dependents
793 1004 FROM dependent
794 1005 WHERE dependent.empno = :input_ssno;
795
796 1006 salary_limit := salary_limit + (count_dependents * 3000);
797
798 1007 The conclusion of the program is straightforward. If the proposed salary is
799 1008 judged as appropriate for this particular employee, then the function result
800 1009 will be 'Salary is valid'. Otherwise, the result will be 'Salary is not valid'.
801 1010 calling program by means of the RETURN statement. If the opposite is true,
802 1011 an immediate termination also occurs.
803 1012
804 1013 IF input_salary > salary_limit THEN
805 1014   RETURN ('Salary is not valid');
806 1015 ELSE
807 1016   RETURN ('Salary is valid');
808 1017 END IF;
809
810 1018 END salary_valid;
811
812 1019 --2
813
814 1020 SET SERVEROUTPUT ON;
Results Explain Describe Saved SQL History
Function created.
0.06 seconds

```

Figure 32: function 1

7.4.2 Execute the function 2b - Test Salary_Valid Function to test output. Provide different values in the test procedure to test each statement in the program logic. For each test statement provide a shot of the code and execution results.

```

SET SERVEROUTPUT ON;

BEGIN

--Test the Salary_Valid function
IF salary_valid('123456789', 80000) THEN
    dbms_output.put_line('Salary is valid');
ELSE
    dbms_output.put_line('Salary is not valid');
END IF;

END;

```

```

103  -- calling program by means of the RETURN(1) statement. If the opposite is true,
104  -- an appropriate termination also occurs.
105
106  -- If input_salary > salary_limit THEN
107      RETURN (FALSE);
108  ELSE
109      RETURN (TRUE);
110  END IF;
111
112  END salary_valid;
113
114  --2
115
116  SET SERVEROUTPUT ON;
117
118  BEGIN
119
120      -- Test the salary_valid function
121      IF salary_valid(923456789, 80000) THEN
122          dbms_output.put_line('Salary is valid');
123      ELSE
124          dbms_output.put_line('Salary is not valid');
125      END IF;
126
127  END;

```

Results Explain Describe Saved SQL History

Salary is not valid
Statement processed.

0.03 seconds

Figure 33: function 2

8 Week 8

- 8.1 Consider the COMPANY database provided in the previous assignments. Using the syntax of Oracle triggers, create triggers to do the following:

Whenever an employee is deleted, delete the PROJECT tuples and DEPENDENT tuples related to that employee, and if the employee manages a department or supervises employees, set the Mgr_ssn for that department to NULL and set the Super_ssn for those employees to NULL.

You are required to provide the trigger code, SQL statements to test the triggers and screenshots of the database state of affected rows before and after executing trigger test statements and an explanation of your logic.

```
CREATE OR REPLACE TRIGGER DELETE_EMPLOYEE
BEFORE DELETE ON EMPLOYEE
FOR EACH ROW

BEGIN

    UPDATE DEPARTMENT
    SET MGRSSN = NULL
    WHERE MGRSSN = :OLD.SSN;

    DELETE
    FROM WORKS_ON
    WHERE ESSN = :OLD.SSN;

    DELETE
    FROM DEPENDENT
    WHERE ESSN = :OLD.SSN;

END;

--TEST
DELETE
FROM EMPLOYEE
WHERE SSN = '888665555';

--confirmation

SELECT *
FROM EMPLOYEE
WHERE SSN = '888665555';

SELECT *
```

```

FROM dependent
WHERE eSSN = '888665555';

SELECT *
FROM works_on
WHERE eSSN = '888665555';

SELECT *
FROM department
WHERE MGRSSN = '888665555';

```

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The right side shows the user 'Jayrup Nakawala' and schema 'WKSP_U2615621'. The main area has tabs for 'Language' (set to SQL) and 'Rows' (set to 10). Below these are buttons for 'Clear Command', 'Find Tables', 'Save', and 'Run'. The code area contains the following PL/SQL trigger creation script:

```

1 CREATE OR REPLACE TRIGGER DELETE_EMPLOYEE
2 BEFORE DELETE ON EMPLOYEE
3 FOR EACH ROW
4
5 BEGIN
6
7   DELETE
8   FROM WORKS ON
9   WHERE ESSN = :OLD.SSN;
10
11  DELETE
12  FROM DEPENDENT
13  WHERE ESSN = :OLD.SSN;
14
15  UPDATE DEPARTMENT
16  SET MGRSSN = NULL
17  WHERE MGRSSN = :OLD.SSN;
18 END;
19

```

The results tab shows the output of the 'Run' command:

```

Trigger created.

0.05 seconds

```

At the bottom, it shows the user 'u2615621@uel.ac.uk', session 'u2615621', and page 'en'. The footer indicates 'Copyright © 1999, 2024, Oracle and/or its affiliates.' and 'Oracle APEX 24.1.3'.

Figure 34: Trigger creation

The screenshot shows the Oracle APEX SQL Workshop interface. In the top navigation bar, 'APEX' is selected, followed by 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The main area is titled 'SQL Commands' with a language dropdown set to 'SQL'. Below the dropdown, there are buttons for 'Rows' (set to 10), 'Clear Command', and 'Find Table'. On the right side, there are 'Save' and 'Run' buttons, along with a user profile icon for 'Jayapriya Nakawala'.

```

1 CREATE OR REPLACE TRIGGER DELETE_EMPLOYEE
2 BEGIN
3   DELETE ON EMPLOYEE
4   FOR EACH ROW
5
6   BEGIN
7     UPDATE DEPARTMENT
8     SET MGRSSN = NULL
9     WHERE MGRSSN = :OLD.SSN;
10
11   DELETE
12     FROM WORKS ON
13     WHERE ESSN = :OLD.SSN;
14
15   DELETE
16     FROM DEPENDENT
17     WHERE ESSN = :OLD.SSN;
18
19   END;
20
21 --TEST
22
23   DELETE
24     FROM EMPLOYEE
25     WHERE SSN = '888665555';
26
27   --confirmation
28
29   SELECT *
30     FROM emp;

```

The 'Results' tab is active, showing the output of the SQL command. The output indicates that the statement was processed in 0.05 seconds. The results table is empty, showing 'no data found'.

Figure 35: Testing the Trigger

This screenshot is identical to Figure 35, showing the same SQL command and its execution results. The 'Results' tab displays the output of the 'SELECT * FROM emp' query, which shows no data found, confirming that the trigger has successfully deleted the employee record.

Figure 36: Confirming that the trigger works

9 Week 9

9.1 What is a Transaction in the context of Databases?

A Transaction is a logical unit of work performed on the database. It takes the database from one consistent state to another consistent state, i.e., duplicate data does not conflict and the Database follows the rules defined.

9.2 What is ACID in Transaction? Define each identified.

ACID stands for:

- **Atomicity:** a transaction is either performed whole or it is not performed at all.
- **Consistency preservation:** a transaction takes the database from one consistent state to another.
- **Isolation:** a transaction is executed independently of other concurrent transactions, ensuring they do not interfere.
- **Durability or permanency:** once a transaction is committed, these changes must never be lost due to subsequent failure.

9.3 What are the different states of a transaction?

These are the following states of a transaction:

- **Active:** this is when the transaction is actively being performed. Read/write occurs at this state.
- **Partially committed:** After the transaction has ended, it enters this state where the transaction still be aborted.
- **Committed:** this state is achieved after successfully committing the transaction.
- **Failed:** When a transaction is aborted, either in the active state or the partially committed state, it enters the failed state.
- **Terminated:** This is the final state where the transaction ends. Which can be achieved from the failed state or the committed state.

9.4 Define COMMIT and ROLLBACK and give example of each.

- **Commit:** This signals that the transaction has been completed successfully so that all the changes/update done to the data item should be saved and safely committed to the database.

- **Rollback:** This signals that the transaction has not been successful and so that any changes/updates done by the transaction should be undone and not saved into the database.

9.5 What are the three main recovery techniques?

- **Immediate Update:** In this technique, the data item is update in the backup as soon as it is modified in the cache.
- **Deferred Update:** In this technique, the modified data item in the cache is updated in the backup after a certain transaction ends or after a certain number of transactions have been completed.
- **Shadow Update:** In this technique, the modified version of a data item does not override the disk copy instead, it is stored in a separate disk.

10 Week 10

10.1 What is the purpose of database security in general?

The main purpose of database security is protecting the database from unauthorized access. The database may contain sensitive information, whether related to personal details or users or projects with strategic importance.

10.2 Describe the approaches for securing DBMSs on the Web.

There are multiple approaches that can be taken to secure DBMSs on the web, some of them are:

- **Privacy:** when transmitting information, it must not be accessible by anyone other than the sender and the receiver.
- **Integrity:** the information should not be changed during transmission
- **Authenticity:** The receiver should be sure that the information came from the sender.

To make sure these practices are ensured, we take help of technologies like proxy servers, firewalls, encryption, SSL certificates, etc.

10.3 Various threats exist on computer system, describe the possible threats and the likelihood of occurring for the followings:

- **Hardware Threats:** Physical damage (fire, water), theft, hardware failure, power issues.
 - Likelihood: Moderate for damage/failure, high for theft in unsecured areas, and moderate for power issues.
- **DBMS and Application Software Threats:** Software bugs, malware, unauthorized access, outdated versions.
 - Likelihood: High for bugs, moderate for malware and access issues, low with updates.
- **Database Threats:** Data breaches, corruption, loss, SQL injection.
 - Likelihood: High for breaches, moderate for loss and corruption, depends on security measures.
- **Communication Networks Threats:** Eavesdropping, DoS, MitM attacks, data leakage.
 - Likelihood: High without encryption, moderate for DoS/MitM, mitigated with security protocols. Users of the System

- **End Users:** Phishing, unintentional errors.
- **Programmers/Operators:** Malicious actions, coding flaws.
- **Admins:** Misuse of privileges, configuration errors.
 - Likelihood: High for user errors, moderate for insider threats.

10.4 What is the purpose of access control matrix? Illustrate your answer with an example.

The Access Control Matrix is used to define and manage the permissions of various users (subjects) to access specific database objects (e.g., tables, files). It maps users to the actions they are allowed to perform (e.g., read, write, update). Example:

Subject/User	Table 1	Table 2
User 1	Read	Read
User 2	Update	Read
User 3	Read, Write	Update

In this example, the matrix defines which operations users can perform on tables, ensuring unauthorized access is prevented.

10.5 Explain how views can be used to improve security of DBMS.

A view is a virtual table that only provides access to the subset of database which the admin allows. Views hide sensitive columns or rows by limiting what the user can see or manipulate, thereby enhancing security.

10.6 What is difference between Discretionary Access Control (DAC) and Mandatory Access Control (MAC)

	Discretionary Access Control (DAC)	Mandatory Access Control (MAC)
Definition	Users have control over their objects and can decide who can access them.	System-enforced access based on predefined policies, not user discretion.
Flexibility:	High; decisions are at the user's discretion.	Low; policies are rigid and centralized.

	Discretionary Access Control (DAC)	Mandatory Access Control (MAC)
Risk:	Susceptible to insider threats due to user control.	More secure, with less reliance on user decisions.
Example:	A user grants or revokes privileges on a table using SQL commands like GRANT and REVOKE.	Classified systems where users can only access data they are cleared for, based on security labels.