Q1 what is trigger?

Ans. A database trigger is a stored procedure that automatically executes whenever an event occurs.

- The event may be insert - update - delete operations.
- Trigger is invoked by Oracle engine automatically whenever a specified event occurs.
- Trigger is stored onto database and invoked repeatedly, when specific condition match.
- Triggers could be defined on the table, view, schema, or database with which the event is associated.

→ Syntax:

```
CREATE [OR REPLACE] TRIGGER trigger-name
      {BEFORE | AFTER | INSTEAD OF}
      {INSERT [OR] | UPDATE [OR] DELETE}
      [OF col_name]
      ON table-name
      [REFERENCING OLD AS o NEW AS n]
      FOR EACH ROW | FOR EACH STATEMENT [WHEN condition]

DECLARE
        declaration-statements

BEGIN
        executable-statements

EXCEPTION
        exception-handling - statements
END;
```

Q2. what are the benefits of Triggers?

Ans. Triggers can have the following benefits:

→ Automation:

- Triggers can automate repetitive tasks, making them easier for developers to complete.

→ Data integrity:

- Triggers can ensure that data is valid and consistent with established standards.

→ Data logging:

- Triggers can create an audit trail of changes made to data

→ Performance:

- Triggers can improve SQL query performance because they don't need to be compiled each time they are executed.

→ Security:

- Triggers can limit exposure of data and logic to authorized users and roles.

→ Consistency:

- Triggers can help ensure data consistency and quality.

→ maintenance:

- Triggers can be easy to maintain

Q3. what are Row triggers and statement triggers?

Ans. 1. Row Trigger

- Row trigger fire for each and every record which are performing INSERT, UPDATE, DELETE from the database table.
- If row deleting is define as trigger event, then trigger is fired, each time row - is deleted from the table.

→ eg:

```
Create or replace trigger check_salary
Before
Insert or update of salary
on copy -EMP
For each row
Begin
    If :new.salary<500 then
        Raise_application_error(-20030, 'minimum salary
                                         is 500');

    End if;
End;
```

2. Statement Trigger

- Statement trigger fire only once for each statement.
- If row deleting is defined as trigger event, then trigger is fired, as all five rows ~~a~~ are deleted from the table.

→ eg:

```
Create a replace trigger dept_check_time
Before
Insert or update or delete
ON departments
Begin
    If to-number (to_char(sysdate, 'hh224')) not
```

```
        between 7 and 15 then
        Raise_application-error (-0010, 'DML operations
                                  not allowed now');

        End if;

End;
```

Q4. ~~what~~ why are we using Before and After triggers?

Ans.

1. BEFORE Trigger:

- BEFORE trigger execute before the triggering DML statement (INSERT, UPDATE, DELETE) execute.
- Triggering SQL statement ~~is~~ may or may not execute, depending on the BEFORE trigger conditions block.

→ eg:

```
CREATE TRIGGER before_insert_occupation
BEFORE INSERT ON employee FOR EACH ROW
BEGIN
        IF NEW.occupation = 'Scientist' THEN SET
              NEW.occupation = 'Doctor';
        END IF;
END;
/
```

2. AFTER Trigger:

- AFTER trigger execute after the triggering DML statement (INSERT, UPDATE, DELETE) executed.
- Triggering SQL statement is execute as soon as followed by the code of trigger before performing Database operation.

- eg:

```
CREATE Trigger after_insert_details
AFTER INSERT ON student_info FOR EACH ROW
BEGIN
INSERT INTO student_detail VALUES (new.stud_id,
                                   new.stud_code,
                    new.stud_name, new.subject, new.marks,
                    new.phone, CURTIME());
END;
/
```

Q5. What is Insert, Update and Delete Triggers?

Ans. 1. INSERT Trigger:

- Insert trigger on mySQL is invoked automatically whenever an insert event occurs on the table
- Syntax:

```
CREATE TRIGGER trigger_name
AFTER INSERT
ON table-name FOR each row
trigge-body;
```

2. UPDATE Trigger:

- The UPDATE trigger in MySQL is invoked automatically whenever an UPDATE event is fired on the table associated with the triggers.
- Syntax:

```
CREATE TRIGGER trigger_name
AFTER UPDATE
ON table-name FOR EACH ROW
trigger-body;
```

3. DELETE Trigger:

- The DELETE trigger in mysql is invoked & automatically whenever a delete event is fired on the table.
- ~~In this~~ Syntax:

```
CREATE TRIGGER trigger_name
UPDATE/DELETE
ON table-name FOR EACH ROW
Trigger_body;
```