Oracle Database 11g

SQL Fundamentals – Lab 4

Table Constraints

Table Constraints

- Constraints enforce rules on the table.
- Constraints prevent the deletion of a table if there are dependencies.
- The Oracle Server uses *constraints* to prevent invalid data entry into tables.

- The following constraint types are valid:
  - `NOT NULL`
  - `UNIQUE`
  - `PRIMARY KEY`
  - `FOREIGN KEY`
  - `CHECK`

Defining Constraints

Constraints can be defined at Create table statement at one of two levels:

- Column constraint level (as part of the column definition)

```
Column datatype [CONSTRAINT constraint_name] constraint_type ,
```

- Table constraint level (as part of the table definition)

```
column,...
  [CONSTRAINT constraint_name] constraint_type
  (column, ...),
```

Column constraint level

```
CREATE TABLE employees(
    employee_id        NUMBER(6)  PRIMARY KEY,
    last_name          VARCHAR2(25)  UNIQUE,
    email              VARCHAR2(25),
    salary          NUMBER(2)
                  CHECK  (salary > 0),
    hire_date          DATE NOT NULL,
   department_id   NUMBER(4)
   REFERENCES   departments   (department_id)  ,
   ... )
```

```
CREATE TABLE employees(
...
hire_date  DATE  CONSTRAINT hire_const NOT NULL,
... )
```

Table constraint level

```
CREATE TABLE employees(
    employee_id        NUMBER(6),
    last_name          VARCHAR2(25)
    ...
    UNIQUE(email)
    PRIMARY KEY(department_id)
);
```

```
CREATE TABLE employees(
    employee_id        NUMBER(6),
    last_name          VARCHAR2(25)
    ...
   CONSTRAINT emp_email_uk UNIQUE(email))
   CONSTRAINT dept_id_pk PRIMARY KEY(department_id)
);
```

Composite key Constraint

- A composite PRIMARY KEY is created by using the table-level definition.
- A composite unique key is created by using the table level definition.
- A table can have only one PRIMARY KEY constraint but can have several UNIQUE constraints.
- A composite foreign key must be created by using the table-level definition.

Two Constraints on Column Level

- To add two constraints on the column level, leave a space between them.

```
CREATE TABLE employees(
    employee_id    NUMBER(6),
    last_name      VARCHAR2(25) NOT NULL  UNIQUE ,
    salary         NUMBER(8,2),
    commission_pct NUMBER(2,2),
    hire_date      DATE
                   CONSTRAINT emp_hire_date_nn
                   NOT NULL,
...
```

Defining Constraints

- Constraints can be defined at Alter table statement

```
ALTER TABLE  table
ADD [CONSTRAINT constraint] type (column);
```

```
ALTER TABLE employees
ADD FOREIGN KEY(department_id)
REFERENCES departments (department_id);
```

FOREIGN KEY Constraint Keywords

- If the data type of the foreign key does not match the data type of the primary key, it will not be created.
- FOREIGN KEY: Defines the column in the child table at the table constraint level
- REFERENCES: Identifies the table and column in the parent table
- ON DELETE CASCADE: Deletes the dependent rows in the child table when a row in the parent table is deleted.
- ON DELETE SET NULL: Converts dependent foreign key values to null
- Without the ON DELETE CASCADE or the ON DELETE SET NULL options, the row in the parent table cannot be deleted if it is referenced in the child table.

Dropping a Constraint

- Remove the manager constraint from the EMPLOYEES table.

```
ALTER  TABLE      employees
DROP CONSTRAINT  emp_manager_fk;
Table altered.
```

- Remove the PRIMARY KEY constraint on the DEPARTMENTS table and drop the associated FOREIGN KEY constraint on the EMPLOYEES.DEPARTMENT_ID column.

```
ALTER  TABLE departments
DROP PRIMARY KEY CASCADE;
Table altered.
```

Sequences

Sequences

- **Automatically generates unique numbers.**
- **Is typically used to create a primary key value.**

- **Creation Syntax:**

```
Create sequence sequence_name
[Start with n]
[Increment by n]
[{Maxvalue n | nomaxvalue}]
[{Cycle | nocycle}];
```

Sequences (cont.)

- **Start with n:**

Specifies the first sequence number to be generated.

- **Increment by n:**

Specifies the interval between sequence numbers

Default increment =1.

Sequences (cont.)

- **Maxvalue n:**

Specifies the maximum value the sequence can generate.

- **Nomaxvalue:**

Specifies a maximum value of 10 power 27.

This is the default option.

Sequences (cont.)

Cycle | Nocycle:

- **Specifies that the sequence continues to generate values after reaching either its maximum value or not.**
- **If nocycle, when it reaches its max value → Error: Sequence exceeds max value and cannot be instantiated.**
- **Nocycle is the default option.**
- **Do not use Cycle option if the sequence will be used to generate primary key values.**

Sequences (cont.)

Example:

```
   create sequence id_seq
    start with 1
   increment by 1
   maxvalue 2000;
```

Nextval: returns the next available sequence value

```
Insert into dept Values  (id_seq.nextval, ’finance’ , ’cairo’);
```

Dropping the sequence:
drop sequence id_seq;

Transaction Control Language (TCL)

Transaction Control Language (TCL)

Transaction Control (TCL) statements are used to manage the changes made by DML statements. It allows statements to be grouped together into logical transactions.

Saving changes

- **Explicitly**
  - **Commit;**
- **Implicitly**
  - when DDL Command or DCL command is issued.
  - a normal exit from SQL Developer without issuing COMMIT or ROLLBACK.

Undo changes

- **Explicitly**
  - **Rollback;**
- **Implicitly**
  - when abnormal termination of SQL Developer or system failure.

State of the Data before COMMIT or ROLLBACK

- The previous state of the data can be recovered because the database buffer is affected.

- The current user can review the result of DML operations by using SELECT statement.

- Other users can’t view the changes made by the current user.

- The affected rows are locked; other users cannot change the data within the affected rows.

State of the Data After COMMIT

- Data changes are written to the database.
- All users can view the results.
- Locks on the affected rows are released; those rows are available for other users to manipulate.

Example (Commit)

```
SQL> INSERT INTO dept
        VALUES  (11, ‘ Finance’, ‘Alex’);
1 row created.

SQL> UPDATE emp
        SET deptno = 40
        WHERE empno = 7902;
1 row updated.

SQL> COMMIT;
Commit complete.
```

State of the Data After ROLLBACK

- Discard all pending changes.
- Data changes are undone.
- Locks on the affected rows are released; those rows are available for other users to manipulate.

Example:

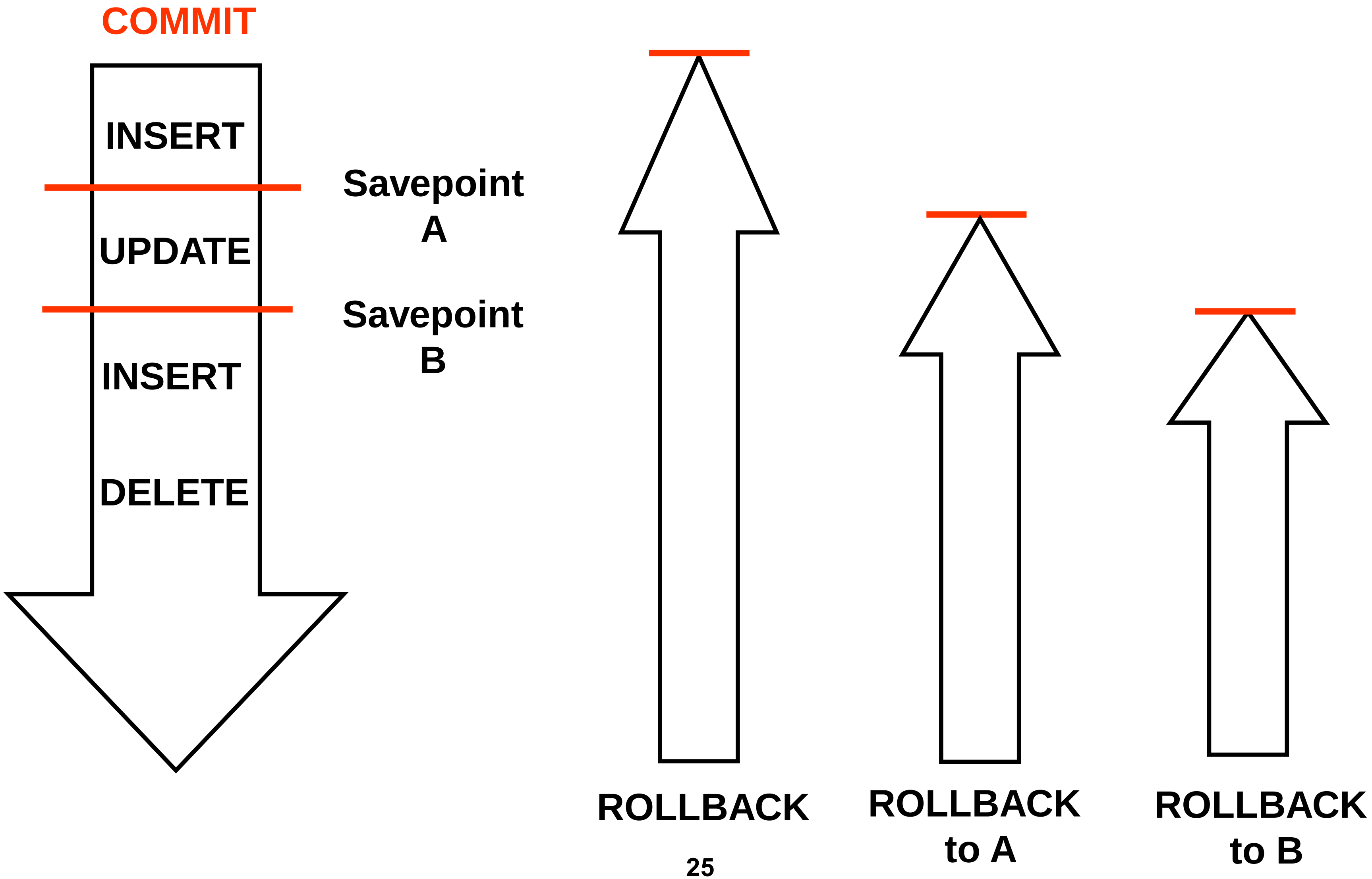
```
SQL> DELETE From test;
25.000 rows deleted.

SQL> ROLLBACK;
Rollback complete.
```

ROLLBACK TO SAVEPOINT

- **The transaction is divided into smaller sections by creating a marker within a current transaction by using the SAVEPOINT command.**

- **Rollback to that marker by using the ROLLBACK TO SAVEPOINT command. It gives the user a chance either to save or rollback code before savepoint.**

ROLLBACK TO SAVEPOINT



COMMIT

INSERT

Savepoint A

UPDATE

Savepoint B

INSERT

DELETE

ROLLBACK

ROLLBACK to A

ROLLBACK to B

Example

```
SQL> UPDATE emp
        SET sal = sal * 1.1
        WHERE job = ‘CLERK’;
10 rows updated.

SQL> SAVEPOINT update_done;
Savepoint created.

SQL> INSERT INTO region
        (id, name)
       VALUES  (8, ‘central’);
1 row inserted.
```

```
SQL>SELECT *
       FROM region
       WHERE id = 8;

         ID     NAME
       -------  -----------
         8         central

SQL> ROLLBACK TO
update_done;
Rollback complete.

SQL>SELECT *
             FROM region
             WHERE id = 8
No rows selected.
```

QUIZ

- you will have a quiz on SQL in your lab
- (week starts at 21-Nov-2020) for half of each section
- (week starts at 28-Nov-2020) for the second half of each section
- Please attend in your section time
- Please bring your laptop if you have successfully installed Oracle 11g Database
- Grade: 5 marks
- Duration: 15 minutes

Good Luck All ☺

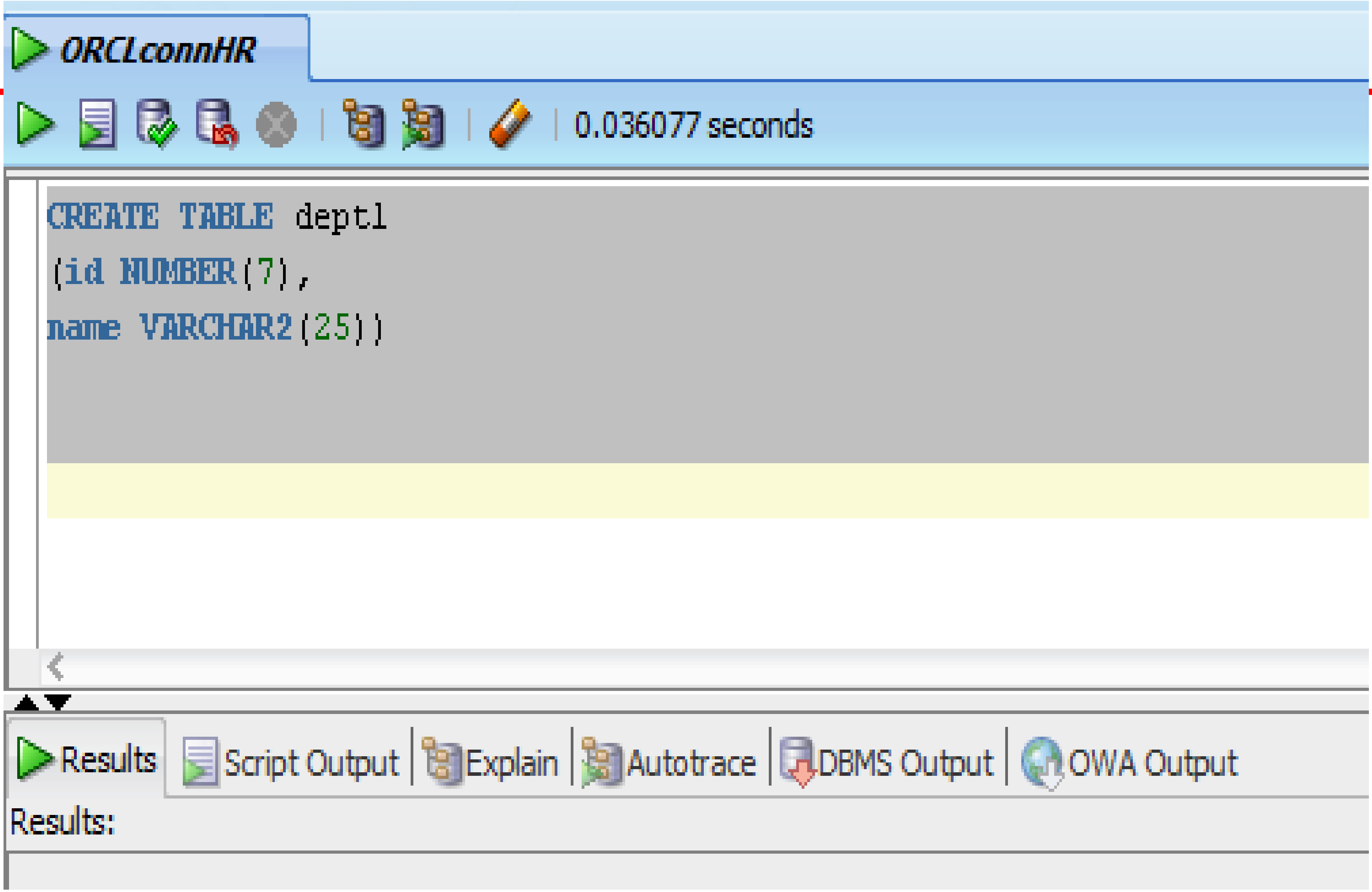
Practice
SQL Developer

Practice

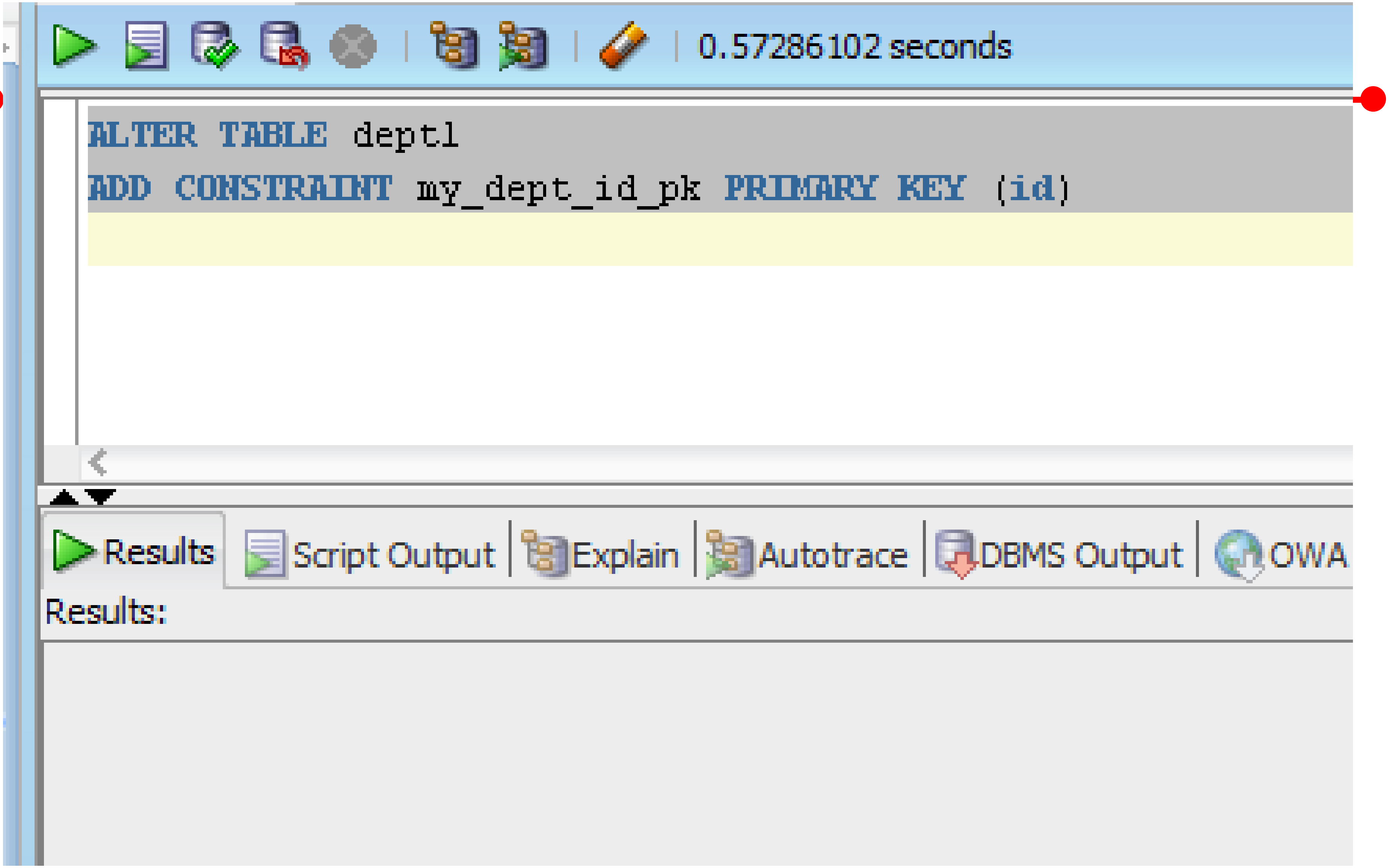
- Create the dept1 table based on the following table instance chart.

Column Name	ID	Name
Data type	NUMBER	VARCHAR2
Length	7	25

- Add a PRIMARY KEY constraint to the dept1 table on the ID column.



```
CREATE TABLE dept1
(id NUMBER(7),
name VARCHAR2(25))
```



```
ALTER TABLE dept1
ADD CONSTRAINT my_dept_id_pk PRIMARY KEY (id)
```

Thank You