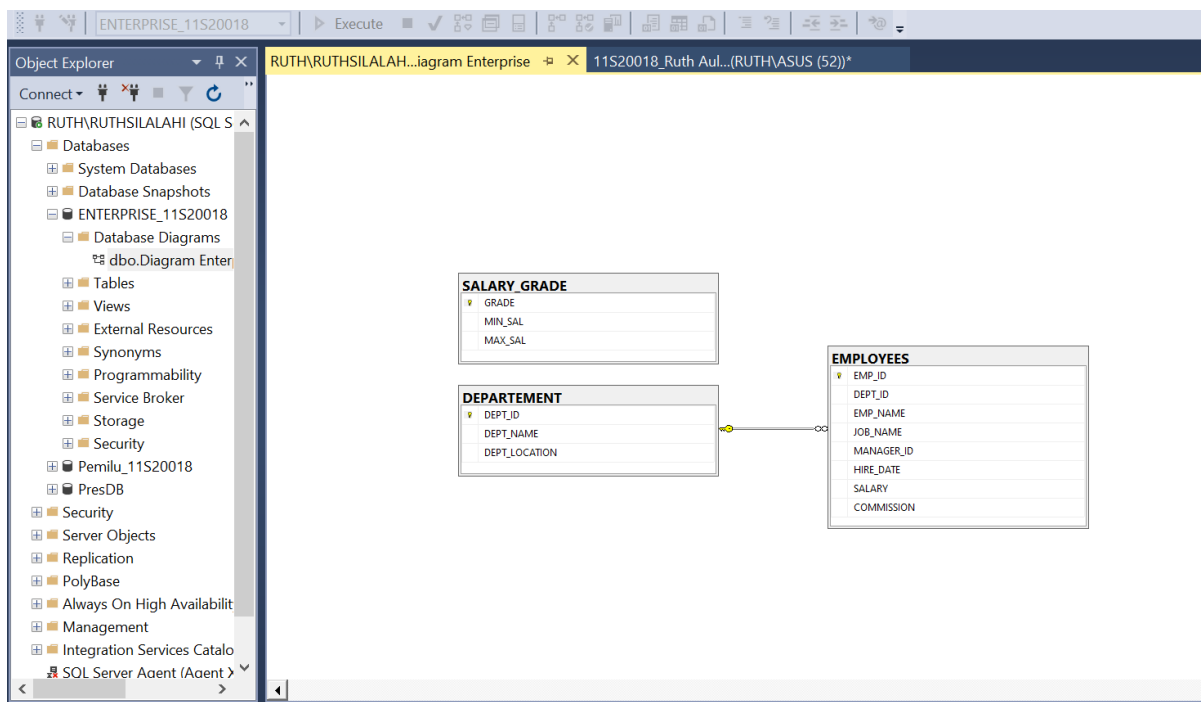
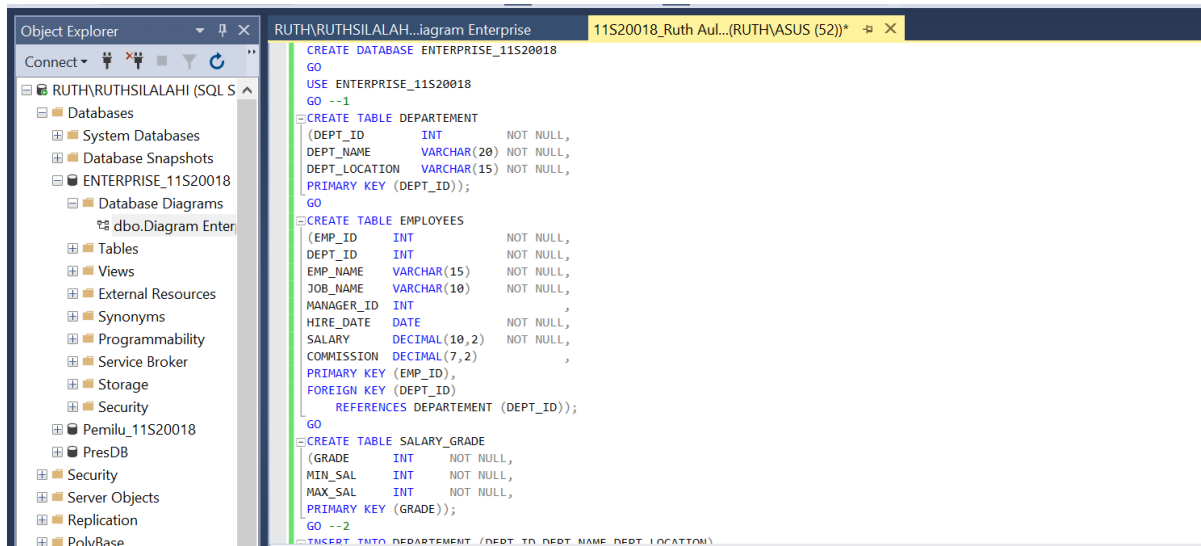


1. Tulislah query untuk membuat table sesuai dengan struktur database pada Gambar 1!



2. Tulislah query untuk mengisi data pada setiap table

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for ENTERPRISE_11S20018. The main window shows the execution of a query in the SQL Server Query Analyzer. The query includes the following statements:

```
MANAGER_ID INT NOT NULL,
HIRE_DATE DATE NOT NULL,
SALARY DECIMAL(10,2) NOT NULL,
COMMISSION DECIMAL(7,2),
PRIMARY KEY (EMP_ID),
FOREIGN KEY (DEPT_ID)
REFERENCES DEPARTEMENT (DEPT_ID));
GO
CREATE TABLE SALARY_GRADE
(
GRADE INT NOT NULL,
MIN_SAL INT NOT NULL,
MAX_SAL INT NOT NULL,
PRIMARY KEY (GRADE));
GO
INSERT INTO DEPARTEMENT (DEPT_ID,DEPT_NAME,DEPT_LOCATION)
VALUES (1001, 'FINANCE', 'SYDNEY'),
(2001, 'AUDIT', 'MELBOURNE'),
(3001, 'MARKETING', 'PERTH'),
(4001, 'PRODUCTION', 'BRISBANE');
SELECT * FROM DEPARTEMENT
GO
INSERT INTO EMPLOYEES (EMP_ID,DEPT_ID,EMP_NAME,JOB_NAME,MANAGER_ID,HIRE_DATE,SALARY,COMMISSION)
VALUES (68319, 1001, 'KAYLING', 'PRESIDENT', NULL, '1991-11-18', 6000.00, NULL),
(69324, 1001, 'MARKER', 'CLERK', 67832, '1992-01-23', 1400.00, NULL);
```

The Results pane shows the output of the SELECT statement, displaying the DEPT_ID, DEPT_NAME, and DEPT_LOCATION for the four departments.

DEPT_ID	DEPT_NAME	DEPT_LOCATION
1	1001	FINANCE
2	2001	AUDIT
3	3001	MARKETING
4	4001	PRODUCTION

Query executed successfully. RUTH\RUTHSILALAH (15.0)

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for ENTERPRISE_11S20018. The main window shows the execution of a query in the SQL Server Query Analyzer. The query includes the following statements:

```
PRIMARY KEY (GRADE));
GO
INSERT INTO DEPARTEMENT (DEPT_ID,DEPT_NAME,DEPT_LOCATION)
VALUES (1001, 'FINANCE', 'SYDNEY'),
(2001, 'AUDIT', 'MELBOURNE'),
(3001, 'MARKETING', 'PERTH'),
(4001, 'PRODUCTION', 'BRISBANE');
SELECT * FROM DEPARTEMENT
GO
INSERT INTO EMPLOYEES (EMP_ID,DEPT_ID,EMP_NAME,JOB_NAME,MANAGER_ID,HIRE_DATE,SALARY,COMMISSION)
VALUES (68319, 1001, 'KAYLING', 'PRESIDENT', NULL, '1991-11-18', 6000.00, NULL),
(69324, 1001, 'MARKER', 'CLERK', 67832, '1992-01-23', 1400.00, NULL),
(67832, 1001, 'CLARE', 'MANAGER', 68319, '1991-06-09', 2550.00, NULL),
(65646, 2001, 'JONAS', 'MANAGER', 68319, '1991-04-02', 2957.00, NULL),
(64989, 3001, 'ADELYN', 'SALESMAN', 66928, '1991-02-20', 1700.00, 400.00),
(65271, 3001, 'WADE', 'SALESMAN', 66928, '1991-02-22', 1350.00, 600.00),
(67858, 2001, 'SCARLET', 'ANALYST', 65646, '1997-04-19', 3100.00, NULL);
SELECT * FROM EMPLOYEES
GO
INSERT INTO SALARY_GRADE (GRADE,MIN_SAL,MAX_SAL)
VALUES (1, 800, 1300);
```

The Results pane shows the output of the SELECT statement, displaying the EMP_ID, DEPT_ID, EMP_NAME, JOB_NAME, MANAGER_ID, HIRE_DATE, SALARY, and COMMISSION for the seven employees.

EMP_ID	DEPT_ID	EMP_NAME	JOB_NAME	MANAGER_ID	HIRE_DATE	SALARY	COMMISSION
1	64989	3001	ADELYN	SALESMAN	66928	1991-02-20	1700.00
2	65271	3001	WADE	SALESMAN	66928	1991-02-22	1350.00
3	65646	2001	JONAS	MANAGER	68319	1991-04-02	2957.00
4	67832	1001	CLARE	MANAGER	68319	1991-06-09	2550.00
5	67858	2001	SCARLET	ANALYST	65646	1997-04-19	3100.00
6	68319	1001	KAYLING	PRESIDENT	NULL	1991-11-18	6000.00
7	69324	1001	MARKER	CLERK	67832	1992-01-23	1400.00

Query executed successfully. RUTH\RUTHSILALAH (15.0)

Object Explorer

Connect

RUTH\RUTHSILALAH (SQL S)

- Databases
 - System Databases
 - Database Snapshots
 - ENTERPRISE_11S20018
 - Database Diagrams
 - dbo.Diagram Enter
 - Tables
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage
 - Security
 - Pemilu_11S20018
 - PresDB
 - Security
 - Server Objects
 - Replication
 - PolyBase
 - Always On High Availabilit
 - Management

RUTH\RUTHSILALAH...igram Enterprise

11S20018_Ruth Aul...(RUTH\ASUS (52))*

```

INSERT INTO EMPLOYEES (EMP_ID,DEPT_ID,EMP_NAME,JOB_NAME,MANAGER_ID,HIRE_DATE,SALARY,COMMISSION)
VALUES (68319, 1001, 'KAYLING', 'PRESIDENT', NULL, '1991-11-18', 6000.00, NULL),
(69324, 1001, 'MARKER', 'CLERK', 67832, '1992-01-23', 1400.00, NULL),
(67832, 1001, 'CLARE', 'MANAGER', 68319, '1991-06-09', 2550.00, NULL),
(65646, 2001, 'JONAS', 'MANAGER', 68319, '1991-04-02', 2957.00, NULL),
(64989, 3001, 'ADELYN', 'SALESMAN', 66928, '1991-02-20', 1700.00, 400.00),
(65271, 3001, 'WADE', 'SALESMAN', 66928, '1991-02-22', 1350.00, 600.00),
(67858, 2001, 'SCARLET', 'ANALYST', 65646, '1997-04-19', 3100.00, NULL);

SELECT * FROM EMPLOYEES
GO
INSERT INTO SALARY_GRADE (GRADE,MIN_SAL,MAX_SAL)
VALUES (1, 800, 1300),
(2, 1301, 1500),
(3, 1501, 2100),
(4, 2101, 3100),
(5, 3101, 9999);

SELECT * FROM SALARY_GRADE
GO
INSERT INTO EMPLOYEES (EMP_ID,DEPT_ID,EMP_NAME,JOB_NAME,MANAGER_ID,HIRE_DATE,SALARY,COMMISSION)
VALUES (68454, 3001, 'TUCKER', 'SALESMAN', 66928, 1991-09-08, 1600.00, NULL);

GO
  
```

83 %

Results Messages

	GRADE	MIN_SAL	MAX_SAL
1	1	800	1300
2	2	1301	1500
3	3	1501	2100
4	4	2101	3100
5	5	3101	9999

3. Tambahlah data berikut ke tabel employees: emp_id: 68454 emp_name: tucker
job_name: salesman manager_id: 66928 hire_date: 1991-09-08 salary: 1600.00
dep_id: 3001

ENTERPRISE_11S20018

Execute

Object Explorer

Connect

RUTH\RUTHSILALAH (SQL S)

Databases

System Databases

Database Snapshots

ENTERPRISE_11S20018

Database Diagrams

dbo.Diagram Enter

Tables

Views

External Resources

Synonyms

Programmability

Service Broker

Storage

Security

Pemilu_11S20018

PresDB

Security

Server Objects

Replication

PolyBase

Always On High Availabilit

Management

Integration Services Catalo

SQL Server Agent (Agent)

RUTH\RUTHSILALAH...iagram Enterprise

11S20018_Ruth Aul...(RUTH\ASUS (52))*

```

(65271, 3001, 'WADE', 'SALESMAN', 66928, '1991-02-22', 1350.00, 600.00),
(67858, 2001, 'SCARLET', 'ANALYST', 65646, '1997-04-19', 3100.00, NULL);
SELECT * FROM EMPLOYEES
GO
INSERT INTO SALARY_GRADE (GRADE,MIN_SAL,MAX_SAL)
VALUES (1, 800, 1300),
(2, 1301, 1500),
(3, 1501, 2100),
(4, 2101, 3100),
(5, 3101, 9999);
SELECT * FROM SALARY_GRADE
GO
INSERT INTO EMPLOYEES (EMP_ID,DEPT_ID,EMP_NAME,JOB_NAME,MANAGER_ID,HIRE_DATE,SALARY,COMMISSION)
VALUES (68454, 3001, 'TUCKER', 'SALESMAN', 66928, '1991-09-08', 1600.00, NULL);
SELECT * FROM EMPLOYEES
GO
UPDATE EMPLOYEES
SET MANAGER_ID = 67832
WHERE EMP_ID = 68319;

```

83 %

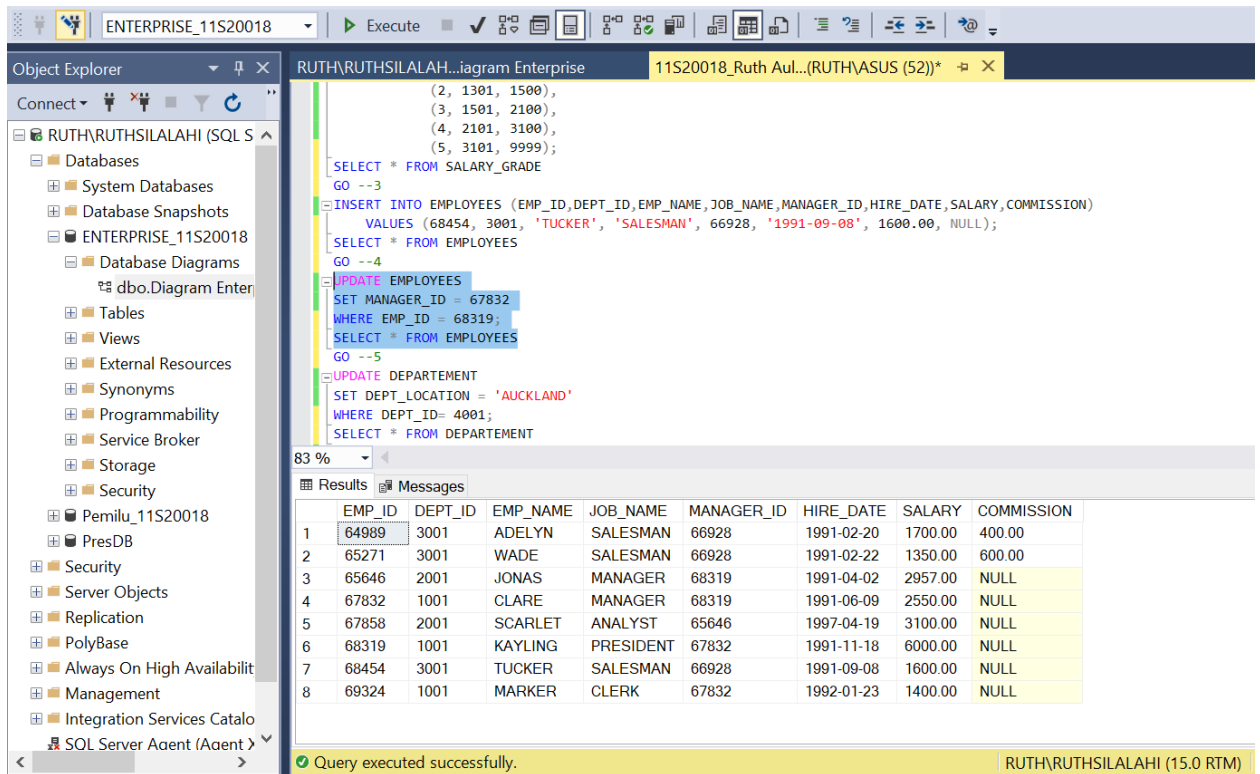
Results Messages

	EMP_ID	DEPT_ID	EMP_NAME	JOB_NAME	MANAGER_ID	HIRE_DATE	SALARY	COMMISSION
1	64989	3001	ADELYN	SALESMAN	66928	1991-02-20	1700.00	400.00
2	65271	3001	WADE	SALESMAN	66928	1991-02-22	1350.00	600.00
3	65646	2001	JONAS	MANAGER	68319	1991-04-02	2957.00	NULL
4	67832	1001	CLARE	MANAGER	68319	1991-06-09	2550.00	NULL
5	67858	2001	SCARLET	ANALYST	65646	1997-04-19	3100.00	NULL
6	68319	1001	KAYLING	PRESIDENT	NULL	1991-11-18	6000.00	NULL
7	68454	3001	TUCKER	SALESMAN	66928	1991-09-08	1600.00	NULL
8	69324	1001	MARKER	CLERK	67832	1992-01-23	1400.00	NULL

Query executed successfully.

RUTH\RUTHSILALAH (15.0 RTM)

4. Tambahlah data manager_id untuk emp_id 68319 dengan 67832



The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The left pane shows the Object Explorer with the database structure of RUTH\RUTHSILALAH (SQL Server). The right pane shows the SQL query editor with the following script:

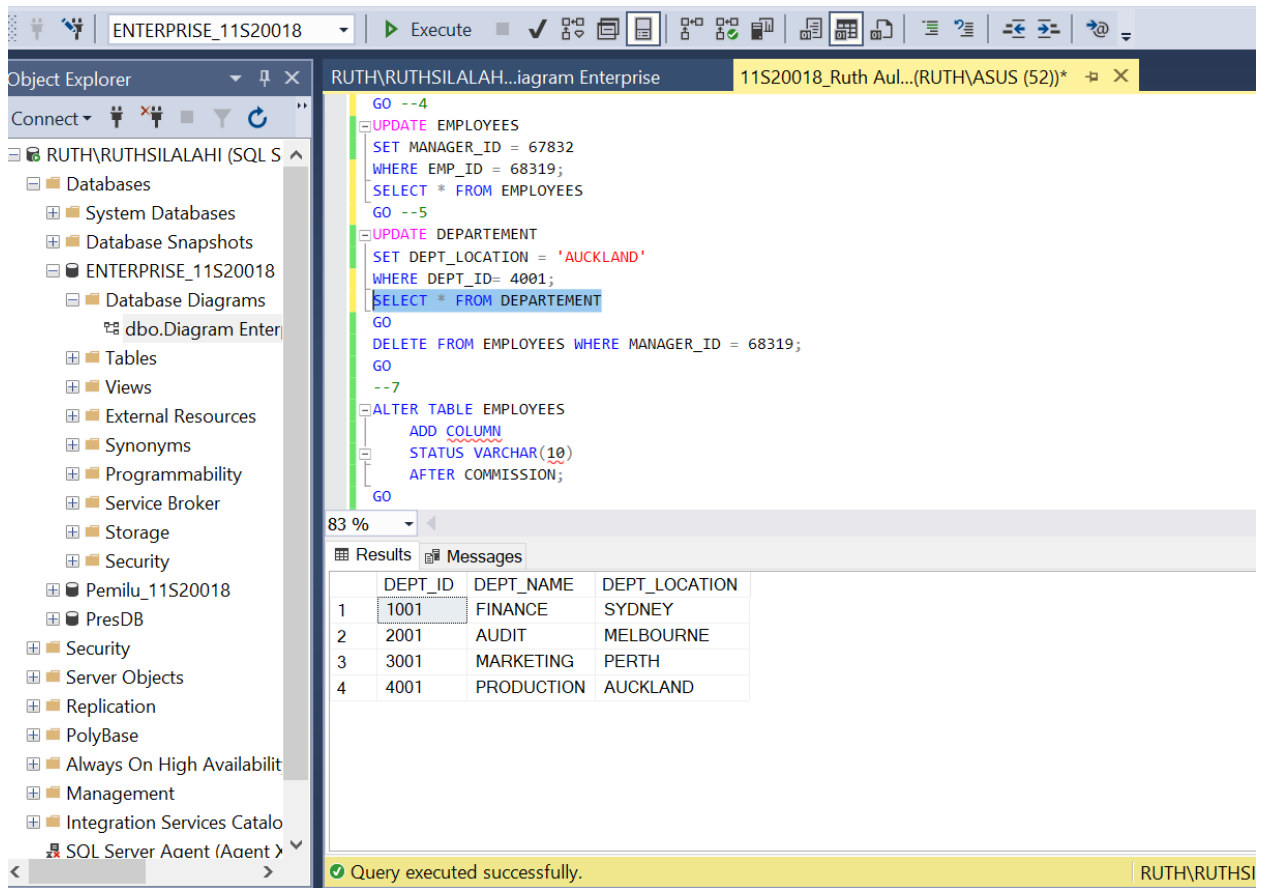
```
(2, 1301, 1500),
(3, 1501, 2100),
(4, 2101, 3100),
(5, 3101, 9999);
SELECT * FROM SALARY_GRADE
GO --3
INSERT INTO EMPLOYEES (EMP_ID,DEPT_ID,EMP_NAME, JOB_NAME,MANAGER_ID,HIRE_DATE,SALARY,COMMISSION)
VALUES (68454, 3001, 'TUCKER', 'SALESMAN', 66928, '1991-09-08', 1600.00, NULL);
SELECT * FROM EMPLOYEES
GO --4
UPDATE EMPLOYEES
SET MANAGER_ID = 67832
WHERE EMP_ID = 68319;
SELECT * FROM EMPLOYEES
GO --5
UPDATE DEPARTEMENT
SET DEPT_LOCATION = 'AUCKLAND'
WHERE DEPT_ID= 4001;
SELECT * FROM DEPARTEMENT
```

The Results pane shows the following data:

	EMP_ID	DEPT_ID	EMP_NAME	JOB_NAME	MANAGER_ID	HIRE_DATE	SALARY	COMMISSION
1	64989	3001	ADELYN	SALESMAN	66928	1991-02-20	1700.00	400.00
2	65271	3001	WADE	SALESMAN	66928	1991-02-22	1350.00	600.00
3	65646	2001	JONAS	MANAGER	68319	1991-04-02	2957.00	NULL
4	67832	1001	CLARE	MANAGER	68319	1991-06-09	2550.00	NULL
5	67858	2001	SCARLET	ANALYST	65646	1997-04-19	3100.00	NULL
6	68319	1001	KAYLING	PRESIDENT	67832	1991-11-18	6000.00	NULL
7	68454	3001	TUCKER	SALESMAN	66928	1991-09-08	1600.00	NULL
8	69324	1001	MARKER	CLERK	67832	1992-01-23	1400.00	NULL

The status bar at the bottom indicates "Query executed successfully." and "RUTH\RUTHSILALAH (15.0 RTM)".

5. Ubahlah data dep_location BRISBANE menjadi AUCKLAND!



The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the Object Explorer for the 'ENTERPRISE_11S20018' database. The right pane shows a query window with the following SQL code:

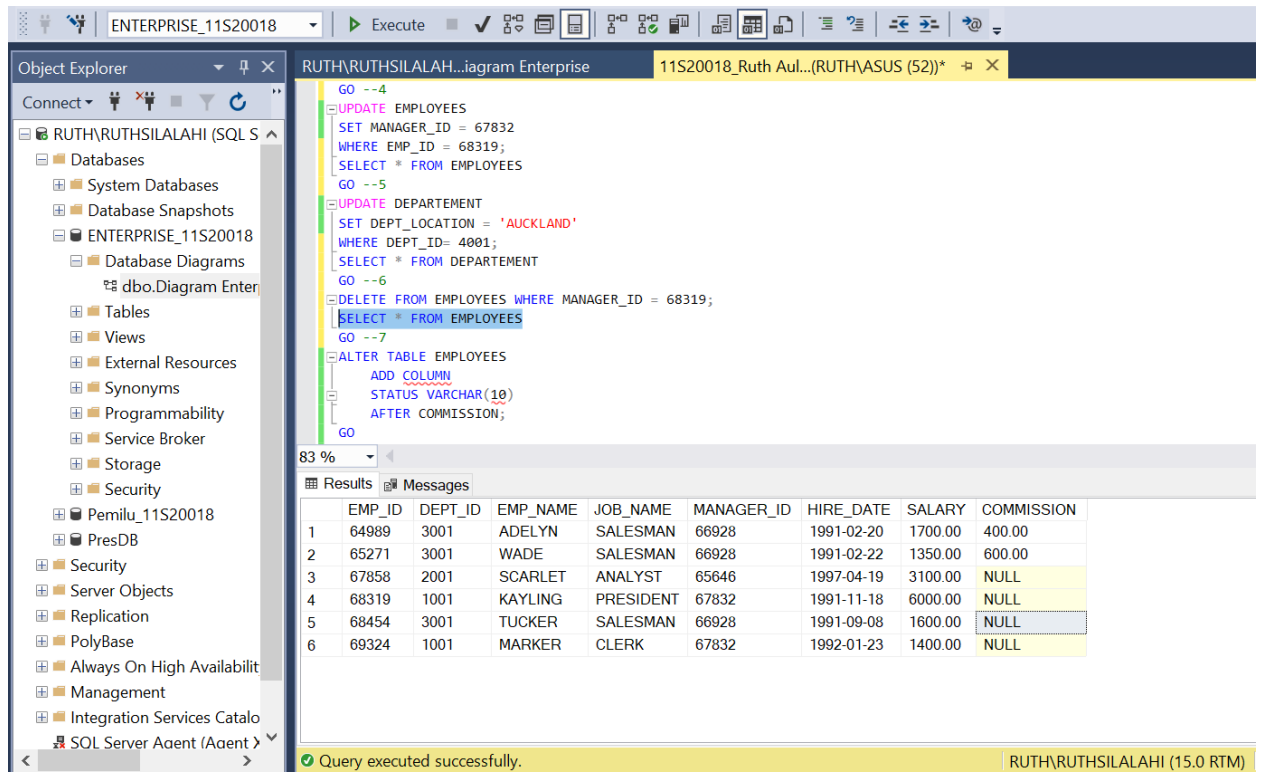
```
GO --4
UPDATE EMPLOYEES
SET MANAGER_ID = 67832
WHERE EMP_ID = 68319;
SELECT * FROM EMPLOYEES
GO --5
UPDATE DEPARTEMENT
SET DEPT_LOCATION = 'AUCKLAND'
WHERE DEPT_ID= 4001;
SELECT * FROM DEPARTEMENT
GO
DELETE FROM EMPLOYEES WHERE MANAGER_ID = 68319;
GO
--7
ALTER TABLE EMPLOYEES
ADD COLUMN
STATUS VARCHAR(10)
AFTER COMMISSION;
GO
```

Below the query window, the 'Results' tab is active, displaying a table with the following data:

	DEPT_ID	DEPT_NAME	DEPT_LOCATION
1	1001	FINANCE	SYDNEY
2	2001	AUDIT	MELBOURNE
3	3001	MARKETING	PERTH
4	4001	PRODUCTION	AUCKLAND

The status bar at the bottom indicates 'Query executed successfully.' and the user 'RUTH\RUTHSI' is logged in.

6. Hapuslah semua data employees yang memiliki manager_id 68319!



The screenshot displays the SQL Server Enterprise Manager interface. The left pane shows the Object Explorer with the database 'ENTERPRISE_11S20018' selected. The right pane shows the 'Query Editor' window with the following SQL script:

```
GO --4
UPDATE EMPLOYEES
SET MANAGER_ID = 67832
WHERE EMP_ID = 68319;
SELECT * FROM EMPLOYEES
GO --5
UPDATE DEPARTEMENT
SET DEPT_LOCATION = 'AUCKLAND'
WHERE DEPT_ID = 4001;
SELECT * FROM DEPARTEMENT
GO --6
DELETE FROM EMPLOYEES WHERE MANAGER_ID = 68319;
SELECT * FROM EMPLOYEES
GO --7
ALTER TABLE EMPLOYEES
ADD COLUMN
STATUS VARCHAR(10)
AFTER COMMISSION;
GO
```

Below the script, the 'Results' tab shows a table with 8 columns: EMP_ID, DEPT_ID, EMP_NAME, JOB_NAME, MANAGER_ID, HIRE_DATE, SALARY, and COMMISSION. The table contains 6 rows of data:

	EMP_ID	DEPT_ID	EMP_NAME	JOB_NAME	MANAGER_ID	HIRE_DATE	SALARY	COMMISSION
1	64989	3001	ADELYN	SALESMAN	66928	1991-02-20	1700.00	400.00
2	65271	3001	WADE	SALESMAN	66928	1991-02-22	1350.00	600.00
3	67858	2001	SCARLET	ANALYST	65646	1997-04-19	3100.00	NULL
4	68319	1001	KAYLING	PRESIDENT	67832	1991-11-18	6000.00	NULL
5	68454	3001	TUCKER	SALESMAN	66928	1991-09-08	1600.00	NULL
6	69324	1001	MARKER	CLERK	67832	1992-01-23	1400.00	NULL

The status bar at the bottom indicates 'Query executed successfully.' and 'RUTH\RUTHSILALAH (15.0 RTM)'.

7. Tambahkan kolom status pada tabel employees yang memiliki tipe data Varchar dengan panjang 10!

The screenshot shows the SQL Enterprise interface with the following SQL query executed:

```

UPDATE DEPARTEMENT
SET DEPT_LOCATION = 'AUCKLAND'
WHERE DEPT_ID= 4001;
SELECT * FROM DEPARTEMENT
GO --6
DELETE FROM EMPLOYEES WHERE MANAGER_ID = 68319;
SELECT * FROM EMPLOYEES
GO --7
ALTER TABLE EMPLOYEES
ADD STATUS VARCHAR(10);
--AFTER COMMISSION;
SELECT * FROM EMPLOYEES
GO
--8
ALTER TABLE EMPLOYESS
DROP COLUMN STATUS;
SELECT * FROM EMPLOYEES
GO
SELECT EMP_NAME, SALARY

```

The Results pane displays the following data:

	EMP_ID	DEPT_ID	EMP_NAME	JOB_NAME	MANAGER_ID	HIRE_DATE	SALARY	COMMISSION	STATUS
1	64989	3001	ADELYN	SALESMAN	66928	1991-02-20	1700.00	400.00	NULL
2	65271	3001	WADE	SALESMAN	66928	1991-02-22	1350.00	600.00	NULL
3	67858	2001	SCARLET	ANALYST	65646	1997-04-19	3100.00	NULL	NULL
4	68319	1001	KAYLING	PRESIDENT	67832	1991-11-18	6000.00	NULL	NULL
5	68454	3001	TUCKER	SALESMAN	66928	1991-09-08	1600.00	NULL	NULL
6	69324	1001	MARKER	CLERK	67832	1992-01-23	1400.00	NULL	NULL

Query executed successfully. RUTH\RUTHSILALAH (15.0 RTM)

8. Hapuslah kolom status pada tabel employees!

The screenshot shows the SQL Enterprise Manager interface. The Object Explorer on the left displays the database structure for RUTH\RUTHSILALAH (SQL S). The central pane shows the execution of a query script. The script includes the following commands:

```
SELECT * FROM DEPARTEMENT
GO --6
DELETE FROM EMPLOYEES WHERE MANAGER_ID = 68319;
SELECT * FROM EMPLOYEES
GO --7
ALTER TABLE EMPLOYEES
ADD STATUS VARCHAR(10);
--AFTER COMMISSION;
SELECT * FROM EMPLOYEES
GO
--8
ALTER TABLE EMPLOYEES
DROP COLUMN STATUS;
SELECT * FROM EMPLOYEES
GO
SELECT EMP_NAME, SALARY
FROM EMPLOYEES;
GO
SELECT DEPT_ID, JOB_NAME
```

The Results pane displays the output of the query, showing a table with 8 columns: EMP_ID, DEPT_ID, EMP_NAME, JOB_NAME, MANAGER_ID, HIRE_DATE, SALARY, and COMMISSION. The data is as follows:

	EMP_ID	DEPT_ID	EMP_NAME	JOB_NAME	MANAGER_ID	HIRE_DATE	SALARY	COMMISSION
1	64989	3001	ADELYN	SALESMAN	66928	1991-02-20	1700.00	400.00
2	65271	3001	WADE	SALESMAN	66928	1991-02-22	1350.00	600.00
3	67858	2001	SCARLET	ANALYST	65646	1997-04-19	3100.00	NULL
4	68319	1001	KAYLING	PRESIDENT	67832	1991-11-18	6000.00	NULL
5	68454	3001	TUCKER	SALESMAN	66928	1991-09-08	1600.00	NULL
6	69324	1001	MARKER	CLERK	67832	1992-01-23	1400.00	NULL

The status bar at the bottom indicates "Query executed successfully." and "RUTH\RUTHSILALAH (15.0 RTM)".

9. Tulislah query untuk menampilkan nama employee beserta gajinya!

The screenshot shows the SQL Server Enterprise interface. The Object Explorer on the left displays the database structure for 'ENTERPRISE_11S20018'. The main window shows a query script with the following SQL commands:

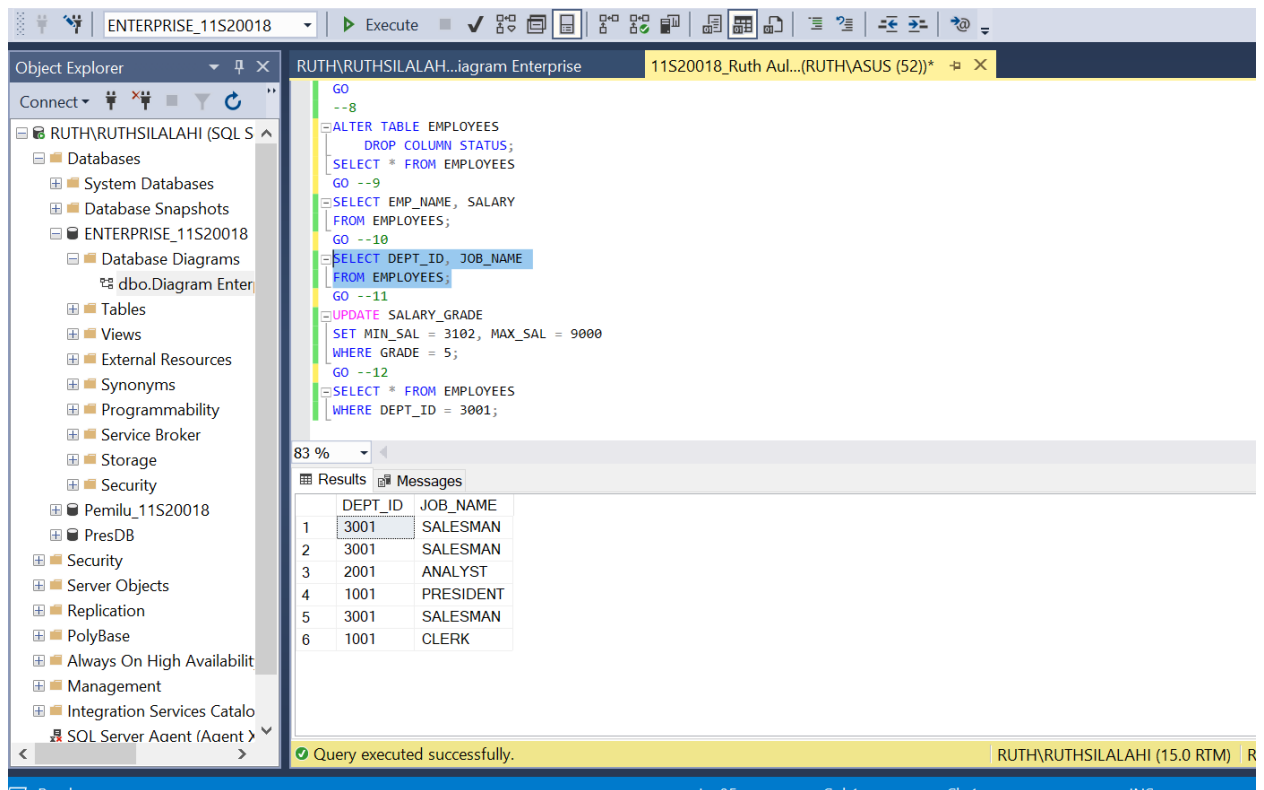
```
GO
--8
ALTER TABLE EMPLOYEES
    DROP COLUMN STATUS;
SELECT * FROM EMPLOYEES
GO --9
SELECT EMP_NAME, SALARY
FROM EMPLOYEES;
GO --10
SELECT DEPT_ID, JOB_NAME
FROM EMPLOYEES;
GO --11
UPDATE SALARY_GRADE
SET MIN_SAL = 3102, MAX_SAL = 9000
WHERE GRADE = 5;
GO --12
SELECT * FROM EMPLOYEES
WHERE DEPT_ID = 3001;
```

The Results pane shows the output of the last query, displaying a table with 6 rows and 2 columns: EMP_NAME and SALARY.

	EMP_NAME	SALARY
1	ADELYN	1700.00
2	WADE	1350.00
3	SCARLET	3100.00
4	KAYLING	6000.00
5	TUCKER	1600.00
6	MARKER	1400.00

The status bar at the bottom indicates 'Query executed successfully.' and 'RUTH\RUTHSILALAH (15.0 RTM)'.

10. Tulislah kueri untuk menampilkan departemen yang unik beserta nama pekerjaan!



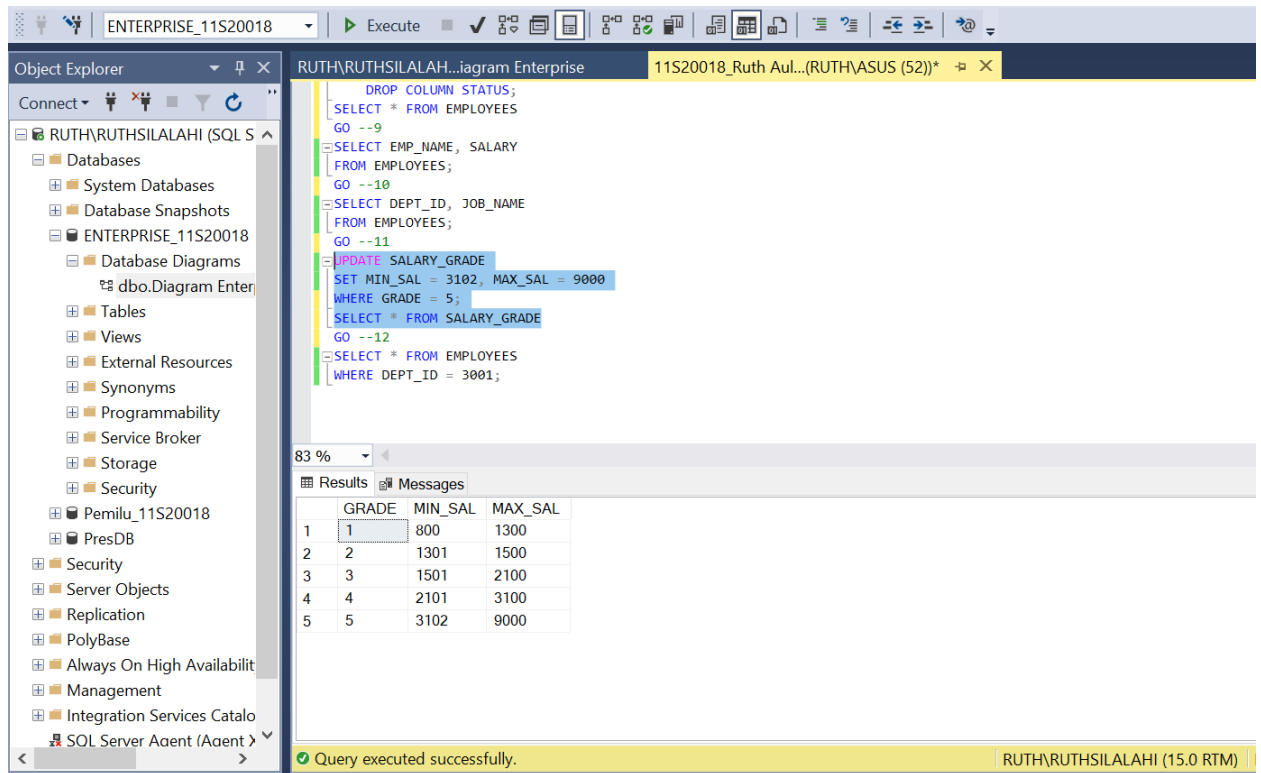
The screenshot displays the SQL Server Enterprise interface. The Object Explorer on the left shows the database structure for 'ENTERPRISE_11S20018'. The central pane shows a SQL query being executed. The query includes several steps: dropping a column, selecting all data, selecting employee names and salaries, selecting unique department IDs and job names, updating salary grades, and finally selecting unique department IDs and job names. The bottom pane shows the results of the final query, which lists 6 rows of unique department IDs and job names.

```
GO
--8
ALTER TABLE EMPLOYEES
    DROP COLUMN STATUS;
SELECT * FROM EMPLOYEES
GO --9
SELECT EMP_NAME, SALARY
FROM EMPLOYEES;
GO --10
SELECT DEPT_ID, JOB_NAME
FROM EMPLOYEES;
GO --11
UPDATE SALARY_GRADE
SET MIN_SAL = 3102, MAX_SAL = 9000
WHERE GRADE = 5;
GO --12
SELECT * FROM EMPLOYEES
WHERE DEPT_ID = 3001;
```

DEPT_ID	JOB_NAME
3001	SALESMAN
3001	SALESMAN
2001	ANALYST
1001	PRESIDENT
3001	SALESMAN
1001	CLERK

Query executed successfully.

11. Ubahlah data minimal salary pada grade 5 menjadi 3102 dan maksimal salary menjadi 9000! Tampilkan semua data salary grade!



The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the Object Explorer with the database 'ENTERPRISE_11S20018' selected. The right pane shows a query window with the following SQL script:

```

DROP COLUMN STATUS;
SELECT * FROM EMPLOYEES
GO --9
SELECT EMP_NAME, SALARY
FROM EMPLOYEES;
GO --10
SELECT DEPT_ID, JOB_NAME
FROM EMPLOYEES;
GO --11
UPDATE SALARY_GRADE
SET MIN_SAL = 3102, MAX_SAL = 9000
WHERE GRADE = 5;
SELECT * FROM SALARY_GRADE
GO --12
SELECT * FROM EMPLOYEES
WHERE DEPT_ID = 3001;

```

The query results are displayed in a table with the following data:

	GRADE	MIN_SAL	MAX_SAL
1	1	800	1300
2	2	1301	1500
3	3	1501	2100
4	4	2101	3100
5	5	3102	9000

The status bar at the bottom indicates 'Query executed successfully.' and 'RUTH\RUTHSILALAH (15.0 RTM)'.

12. . Tampilkan semua data employee yang memiliki department id 3001!

The screenshot displays the SQL Server Enterprise Manager interface. The left pane shows the Object Explorer with the database 'ENTERPRISE_11S20018' selected. The right pane shows the query editor with the following SQL script:

```

-- DROP COLUMN STATUS;
SELECT * FROM EMPLOYEES
GO --9
SELECT EMP_NAME, SALARY
FROM EMPLOYEES;
GO --10
SELECT DEPT_ID, JOB_NAME
FROM EMPLOYEES;
GO --11
UPDATE SALARY_GRADE
SET MIN_SAL = 3102, MAX_SAL = 9000
WHERE GRADE = 5;
SELECT * FROM SALARY_GRADE
GO --12
SELECT * FROM EMPLOYEES
WHERE DEPT_ID = 3001;

```

The query results are displayed in a table with the following columns: EMP_ID, DEPT_ID, EMP_NAME, JOB_NAME, MANAGER_ID, HIRE_DATE, SALARY, and COMMISSION. The results show three employees with DEPT_ID 3001:

EMP_ID	DEPT_ID	EMP_NAME	JOB_NAME	MANAGER_ID	HIRE_DATE	SALARY	COMMISSION
64989	3001	ADELYN	SALESMAN	66928	1991-02-20	1700.00	400.00
65271	3001	WADE	SALESMAN	66928	1991-02-22	1350.00	600.00
68454	3001	TUCKER	SALESMAN	66928	1991-09-08	1600.00	NULL

The status bar at the bottom indicates 'Query executed successfully.' and 'RUTH\RUTHSILALAH (15.0 RTM)'.