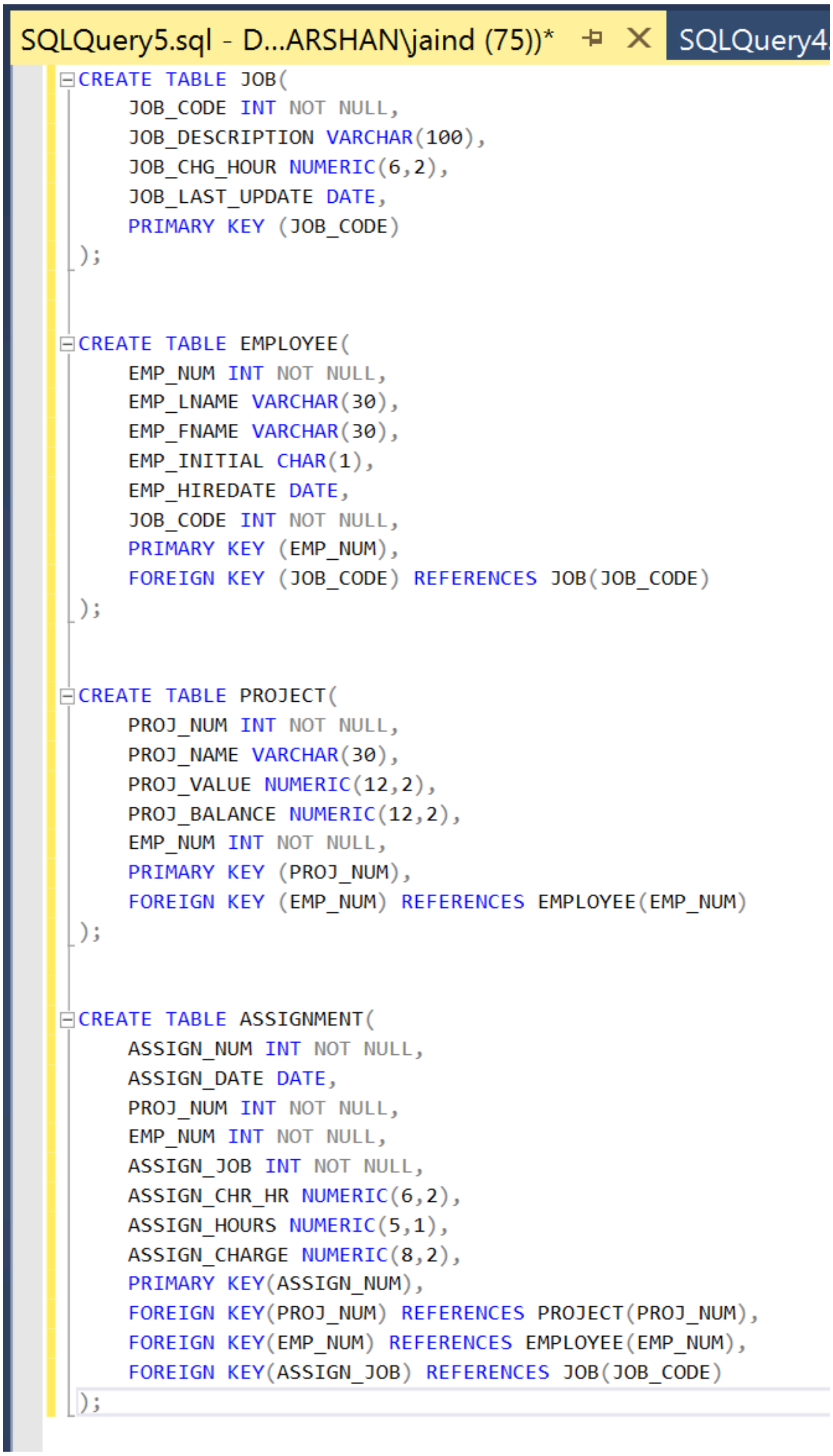
Task 1:



```
CREATE TABLE JOB(
    JOB_CODE INT NOT NULL,
    JOB_DESCRIPTION VARCHAR(100),
    JOB_CHG_HOUR NUMERIC(6,2),
    JOB_LAST_UPDATE DATE,
    PRIMARY KEY (JOB_CODE)
);


CREATE TABLE EMPLOYEE(
    EMP_NUM INT NOT NULL,
    EMP_LNAME VARCHAR(30),
    EMP_FNAME VARCHAR(30),
    EMP_INITIAL CHAR(1),
    EMP_HIREDATE DATE,
    JOB_CODE INT NOT NULL,
    PRIMARY KEY (EMP_NUM),
    FOREIGN KEY (JOB_CODE) REFERENCES JOB(JOB_CODE)
);


CREATE TABLE PROJECT(
    PROJ_NUM INT NOT NULL,
    PROJ_NAME VARCHAR(30),
    PROJ_VALUE NUMERIC(12,2),
    PROJ_BALANCE NUMERIC(12,2),
    EMP_NUM INT NOT NULL,
    PRIMARY KEY (PROJ_NUM),
    FOREIGN KEY (EMP_NUM) REFERENCES EMPLOYEE(EMP_NUM)
);


CREATE TABLE ASSIGNMENT(
    ASSIGN_NUM INT NOT NULL,
    ASSIGN_DATE DATE,
    PROJ_NUM INT NOT NULL,
    EMP_NUM INT NOT NULL,
    ASSIGN_JOB INT NOT NULL,
    ASSIGN_CHR_HR NUMERIC(6,2),
    ASSIGN_HOURS NUMERIC(5,1),
    ASSIGN_CHARGE NUMERIC(8,2),
    PRIMARY KEY(ASSIGN_NUM),
    FOREIGN KEY(PROJ_NUM) REFERENCES PROJECT(PROJ_NUM),
    FOREIGN KEY(EMP_NUM) REFERENCES EMPLOYEE(EMP_NUM),
    FOREIGN KEY(ASSIGN_JOB) REFERENCES JOB(JOB_CODE)
);
```

Task 2:

Entity Integrity:

- Entity integrity ensures that each table has a unique identifier as primary key.
- Each table has a unique identifier such as:
  JOB_CODE for JOB Table.
  EMP_NUM for EMPLOYEE Table.
  PROJ_NUM for PROJECT Table.
  ASSIGN_NUM for ASSIGNMENT Table.

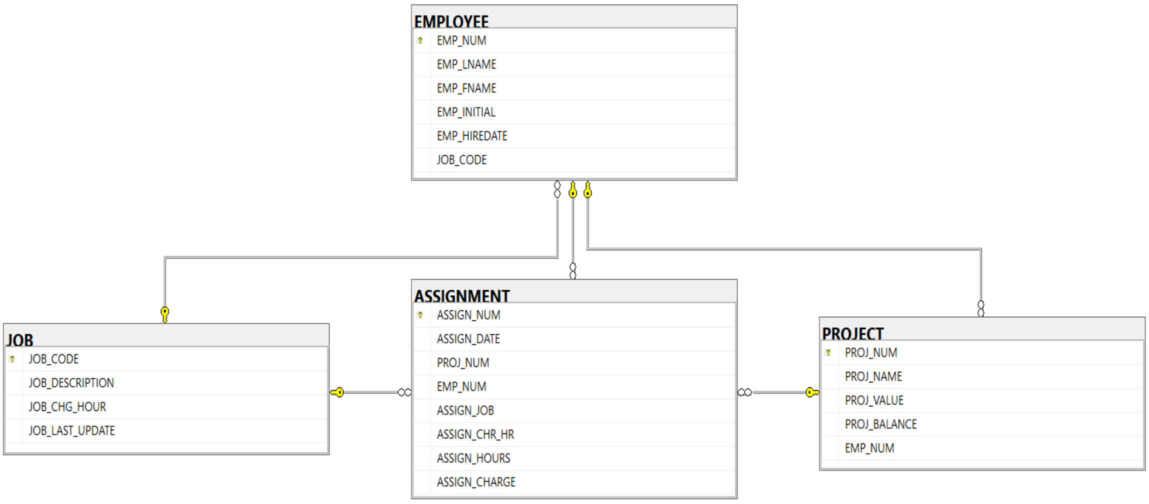
Reference Integrity:

- Reference integrity ensures valid relationships between tables by connecting them with Foreign Keys.
- In the EMPLOYEE Table: JOB_CODE is a foreign key referencing JOB.JOB_CODE. Ensures valid job assignment for each employee.
- In the PROJECT Table: EMP_NUM is a foreign key referencing EMPLOYEE.EMP_NUM. Ensures validity of the employee managing the project.
- In the ASSIGNMENT Table: PROJ_NUM is a foreign key referencing PROJECT.PROJ_NUM, EMP_NUM is a foreign key referencing EMPLOYEE.EMP_NUM and ASSIGN_JOB is a foreign key referencing JOB.JOB_CODE. Ensures validity of existing projects, employees and job roles.
- This ensures that if any data needs to be removed from a table, the data from the table referencing it with a foreign key should be removed first.

Domain Integrity:

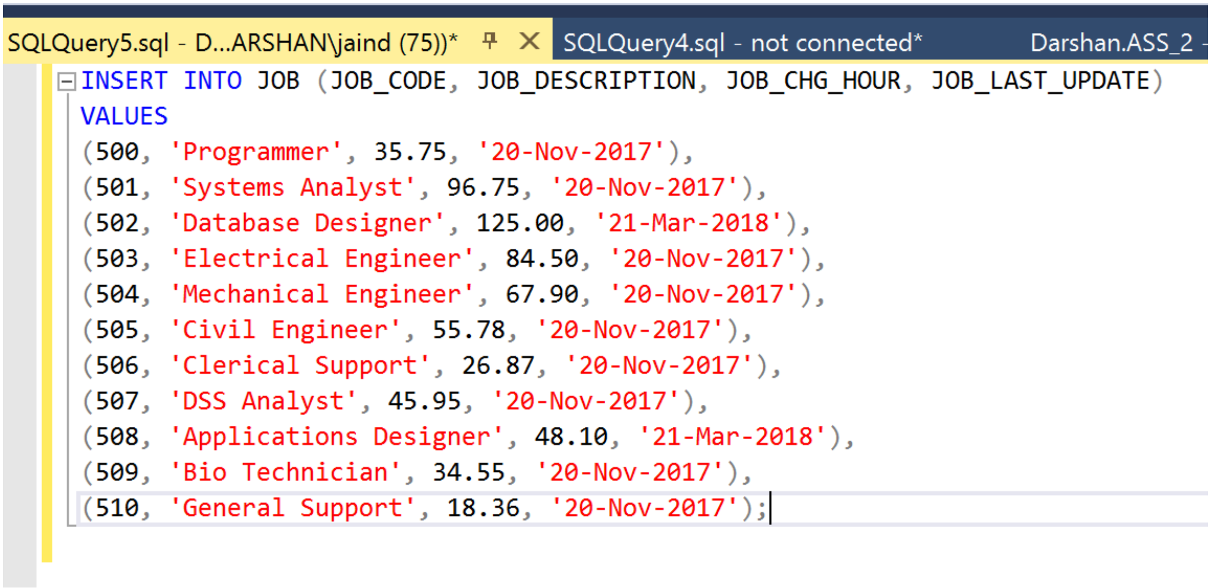
- Domain integrity ensures that columns have appropriate data types and constraints to maintain data validity.
- There is a DATE constraint on ASSIGN_DATE, EMP_HIREDATE AND JOB_LAST_UPDATE. This ensures only valid dates are entered.
- There is a NUMERIC constraint on JOB_CHG_HOUR(6,2) that ensures the maximum numbers should be 6 with 2 digits after the decimal, ASSIGN_HOURS(5,1) that ensures that the maximum numbers should be 5 with just 1 digit after decimal and ASSIGN_CHARGE(8,2) that ensures that the maximum numbers should be 8 with 2 digits after decimal.
- NUMERIC(5,1) CHECK (ASSIGN_HOURS > 0) prevents negative values.

DATABASE DIAGRAM:

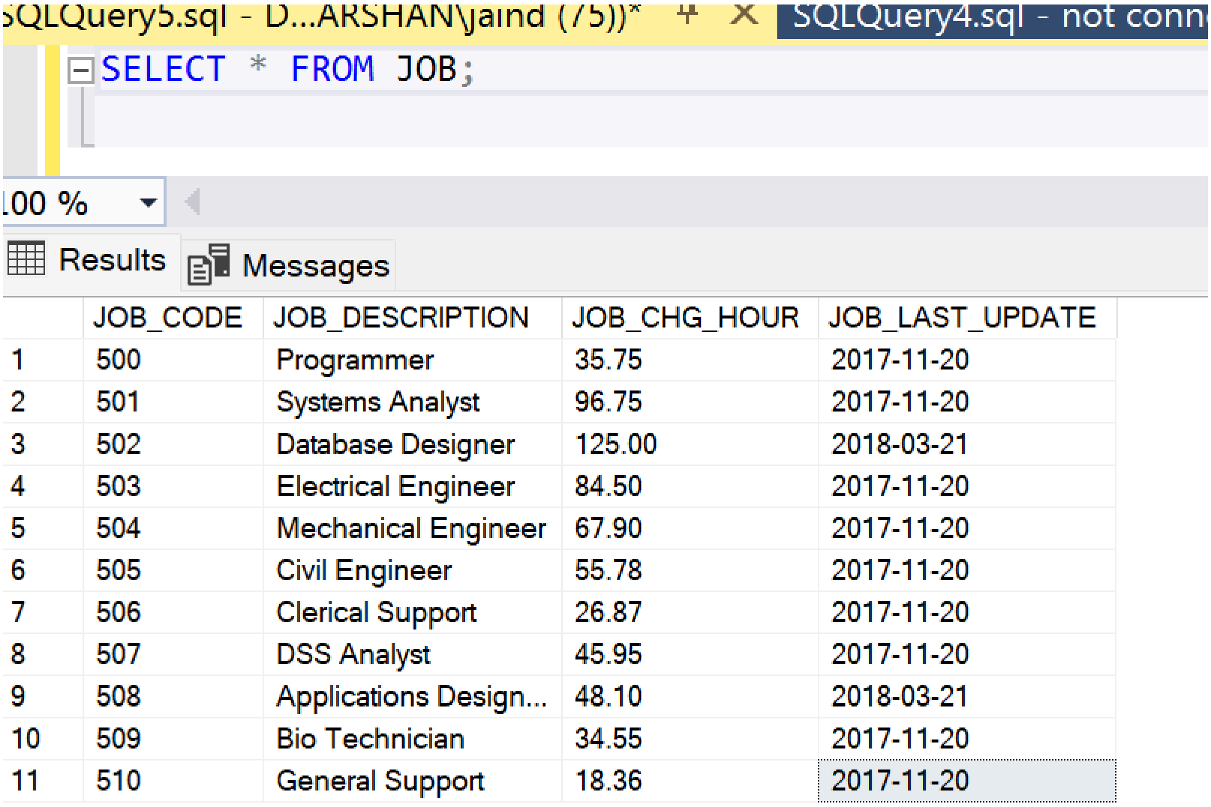


Task 3:

Job Table:


SQLQuery5.sql - D...ARSHAN\jaind (75))*
SQLQuery4.sql - not connected*
Darshan.ASS_2 -

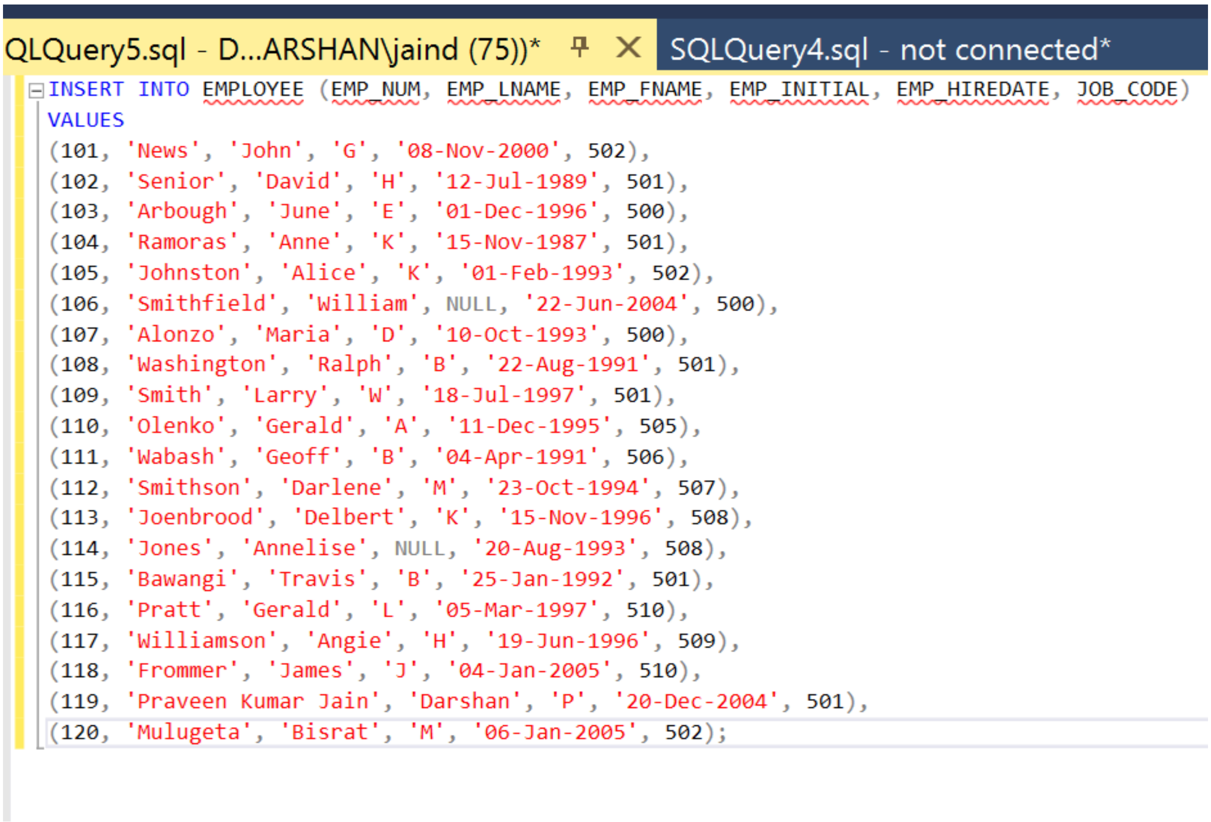
```sql
INSERT INTO JOB (JOB_CODE, JOB_DESCRIPTION, JOB_CHG_HOUR, JOB_LAST_UPDATE)
VALUES
(500, 'Programmer', 35.75, '20-Nov-2017'),
(501, 'Systems Analyst', 96.75, '20-Nov-2017'),
(502, 'Database Designer', 125.00, '21-Mar-2018'),
(503, 'Electrical Engineer', 84.50, '20-Nov-2017'),
(504, 'Mechanical Engineer', 67.90, '20-Nov-2017'),
(505, 'Civil Engineer', 55.78, '20-Nov-2017'),
(506, 'Clerical Support', 26.87, '20-Nov-2017'),
(507, 'DSS Analyst', 45.95, '20-Nov-2017'),
(508, 'Applications Designer', 48.10, '21-Mar-2018'),
(509, 'Bio Technician', 34.55, '20-Nov-2017'),
(510, 'General Support', 18.36, '20-Nov-2017');
```



```sql
SELECT * FROM JOB;
```

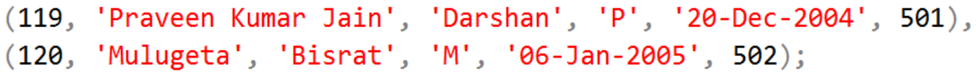
	JOB_CODE	JOB_DESCRIPTION	JOB_CHG_HOUR	JOB_LAST_UPDATE
1	500	Programmer	35.75	2017-11-20
2	501	Systems Analyst	96.75	2017-11-20
3	502	Database Designer	125.00	2018-03-21
4	503	Electrical Engineer	84.50	2017-11-20
5	504	Mechanical Engineer	67.90	2017-11-20
6	505	Civil Engineer	55.78	2017-11-20
7	506	Clerical Support	26.87	2017-11-20
8	507	DSS Analyst	45.95	2017-11-20
9	508	Applications Design...	48.10	2018-03-21
10	509	Bio Technician	34.55	2017-11-20
11	510	General Support	18.36	2017-11-20

Employee Table:

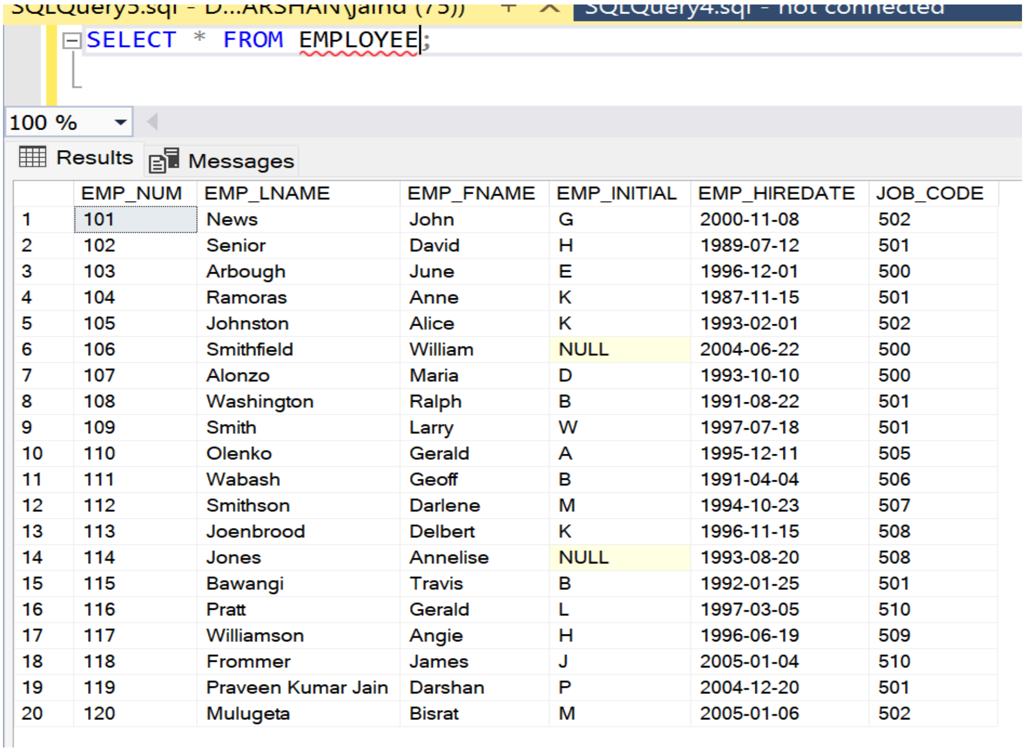

QLQuery5.sql - D...ARSHAN\jaind (75))*
SQLQuery4.sql - not connected*

```
INSERT INTO EMPLOYEE (EMP_NUM, EMP_LNAME, EMP_FNAME, EMP_INITIAL, EMP_HIREDATE, JOB_CODE)
VALUES
(101, 'News', 'John', 'G', '08-Nov-2000', 502),
(102, 'Senior', 'David', 'H', '12-Jul-1989', 501),
(103, 'Arbough', 'June', 'E', '01-Dec-1996', 500),
(104, 'Ramoras', 'Anne', 'K', '15-Nov-1987', 501),
(105, 'Johnston', 'Alice', 'K', '01-Feb-1993', 502),
(106, 'Smithfield', 'William', NULL, '22-Jun-2004', 500),
(107, 'Alonzo', 'Maria', 'D', '10-Oct-1993', 500),
(108, 'Washington', 'Ralph', 'B', '22-Aug-1991', 501),
(109, 'Smith', 'Larry', 'W', '18-Jul-1997', 501),
(110, 'Olenko', 'Gerald', 'A', '11-Dec-1995', 505),
(111, 'Wabash', 'Geoff', 'B', '04-Apr-1991', 506),
(112, 'Smithson', 'Darlene', 'M', '23-Oct-1994', 507),
(113, 'Joenbrood', 'Delbert', 'K', '15-Nov-1996', 508),
(114, 'Jones', 'Annelise', NULL, '20-Aug-1993', 508),
(115, 'Bawangi', 'Travis', 'B', '25-Jan-1992', 501),
(116, 'Pratt', 'Gerald', 'L', '05-Mar-1997', 510),
(117, 'Williamson', 'Angie', 'H', '19-Jun-1996', 509),
(118, 'Frommer', 'James', 'J', '04-Jan-2005', 510),
(119, 'Praveen Kumar Jain', 'Darshan', 'P', '20-Dec-2004', 501),
(120, 'Mulugeta', 'Bisrat', 'M', '06-Jan-2005', 502);
```

The last two rows of Employee table are added with student names as shown below:



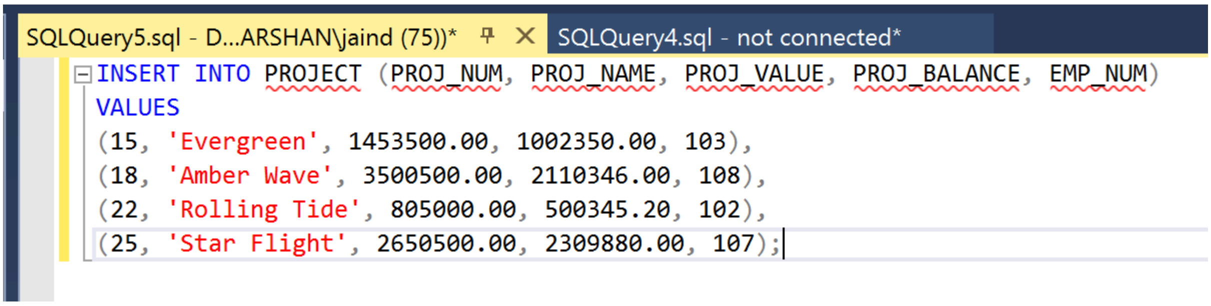
```
(119, 'Praveen Kumar Jain', 'Darshan', 'P', '20-Dec-2004', 501),
(120, 'Mulugeta', 'Bisrat', 'M', '06-Jan-2005', 502);
```


SQLQuery5.sql - D...ARSHAN\jaind (75))
SQLQuery4.sql - not connected
100 %
Results
Messages

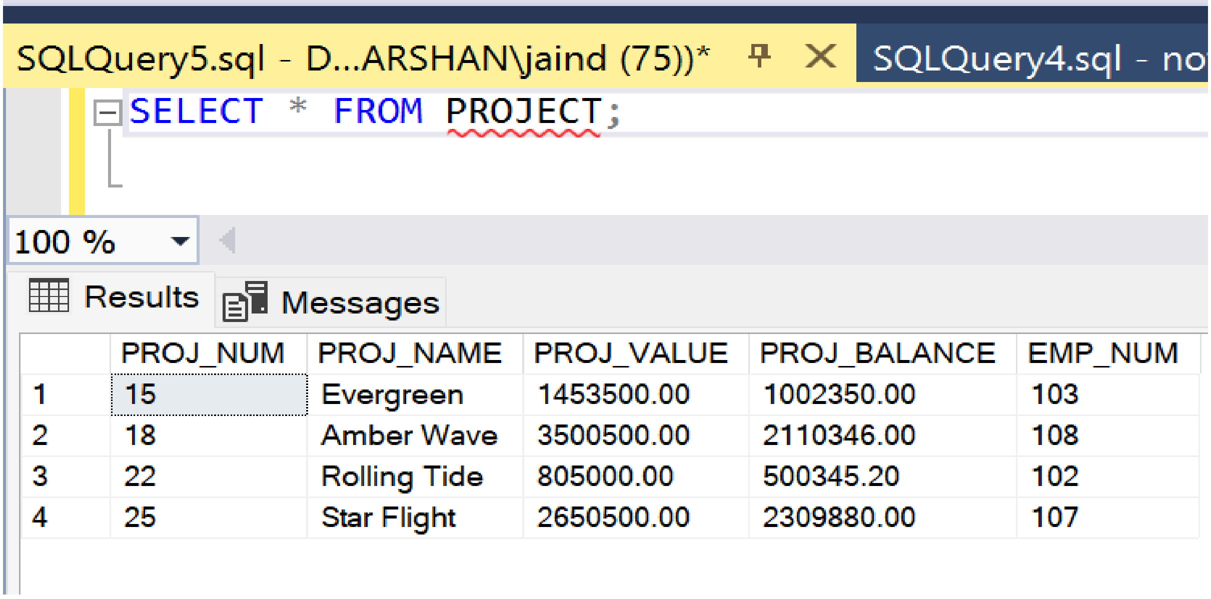
```
SELECT * FROM EMPLOYEE;
```

	EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_HIREDATE	JOB_CODE
1	101	News	John	G	2000-11-08	502
2	102	Senior	David	H	1989-07-12	501
3	103	Arbough	June	E	1996-12-01	500
4	104	Ramoras	Anne	K	1987-11-15	501
5	105	Johnston	Alice	K	1993-02-01	502
6	106	Smithfield	William	NULL	2004-06-22	500
7	107	Alonzo	Maria	D	1993-10-10	500
8	108	Washington	Ralph	B	1991-08-22	501
9	109	Smith	Larry	W	1997-07-18	501
10	110	Olenko	Gerald	A	1995-12-11	505
11	111	Wabash	Geoff	B	1991-04-04	506
12	112	Smithson	Darlene	M	1994-10-23	507
13	113	Joenbrood	Delbert	K	1996-11-15	508
14	114	Jones	Annelise	NULL	1993-08-20	508
15	115	Bawangi	Travis	B	1992-01-25	501
16	116	Pratt	Gerald	L	1997-03-05	510
17	117	Williamson	Angie	H	1996-06-19	509
18	118	Frommer	James	J	2005-01-04	510
19	119	Praveen Kumar Jain	Darshan	P	2004-12-20	501
20	120	Mulugeta	Bisrat	M	2005-01-06	502

Project Table:


SQLQuery5.sql - D...ARSHAN\jaind (75))*
SQLQuery4.sql - not connected*

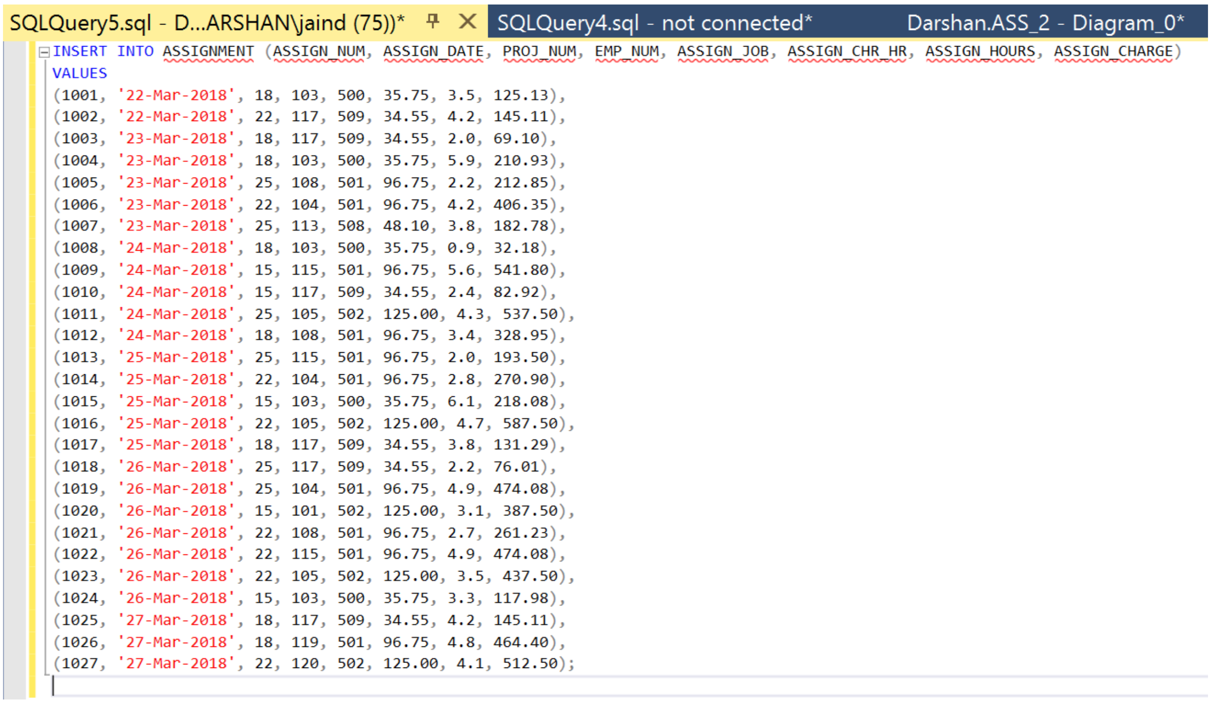
```sql
INSERT INTO PROJECT (PROJ_NUM, PROJ_NAME, PROJ_VALUE, PROJ_BALANCE, EMP_NUM)
VALUES
(15, 'Evergreen', 1453500.00, 1002350.00, 103),
(18, 'Amber Wave', 3500500.00, 2110346.00, 108),
(22, 'Rolling Tide', 805000.00, 500345.20, 102),
(25, 'Star Flight', 2650500.00, 2309880.00, 107);
```



```sql
SELECT * FROM PROJECT;
```

	PROJ_NUM	PROJ_NAME	PROJ_VALUE	PROJ_BALANCE	EMP_NUM
1	15	Evergreen	1453500.00	1002350.00	103
2	18	Amber Wave	3500500.00	2110346.00	108
3	22	Rolling Tide	805000.00	500345.20	102
4	25	Star Flight	2650500.00	2309880.00	107

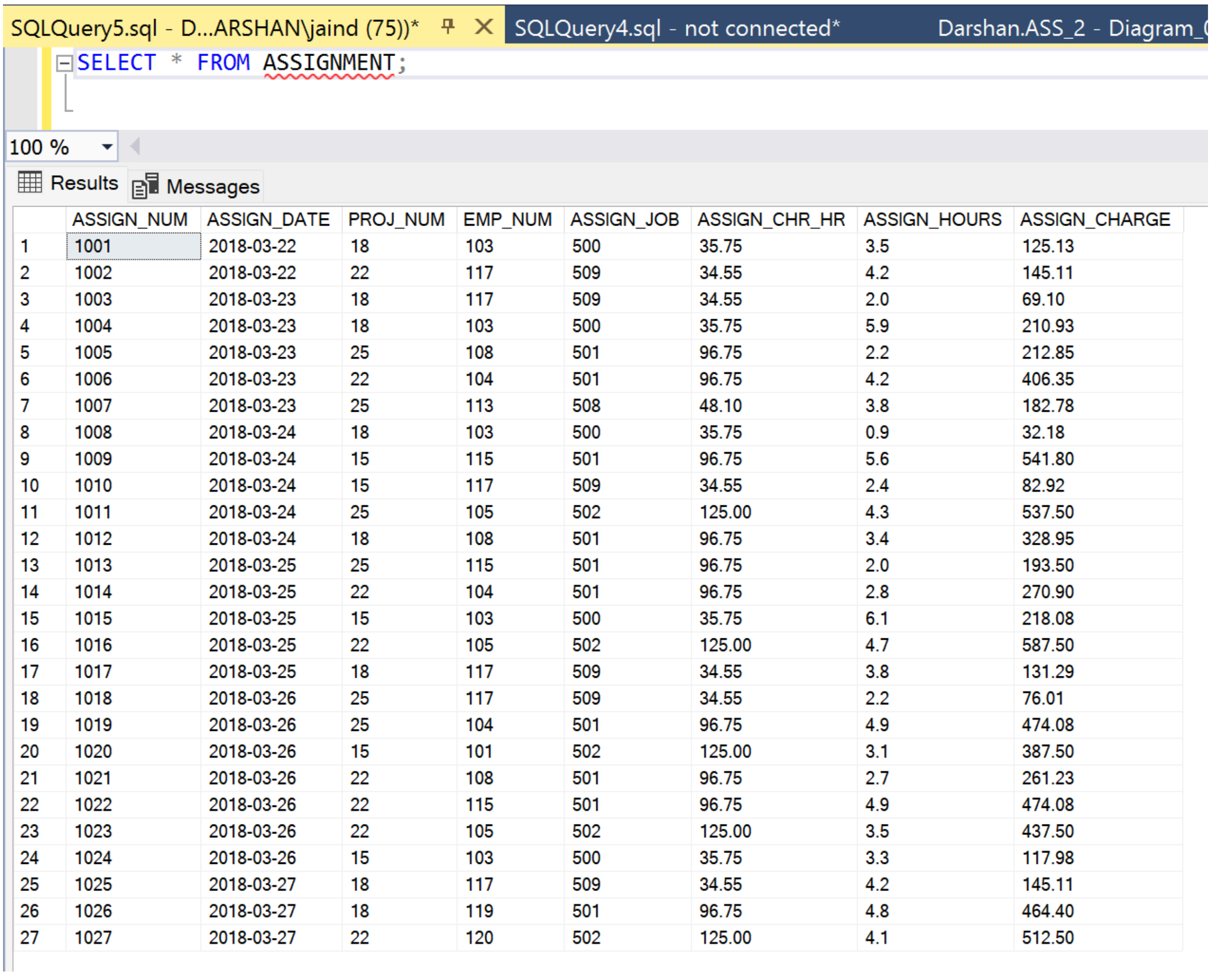
Assignment Table:


SQLQuery5.sql - D...ARSHAN\jaind (75))*
SQLQuery4.sql - not connected*
Darshan.ASS_2 - Diagram_0*

```sql
INSERT INTO ASSIGNMENT (ASSIGN_NUM, ASSIGN_DATE, PROJ_NUM, EMP_NUM, ASSIGN_JOB, ASSIGN_CHR_HR, ASSIGN_HOURS, ASSIGN_CHARGE)
VALUES
(1001, '22-Mar-2018', 18, 103, 500, 35.75, 3.5, 125.13),
(1002, '22-Mar-2018', 22, 117, 509, 34.55, 4.2, 145.11),
(1003, '23-Mar-2018', 18, 117, 509, 34.55, 2.0, 69.10),
(1004, '23-Mar-2018', 18, 103, 500, 35.75, 5.9, 210.93),
(1005, '23-Mar-2018', 25, 108, 501, 96.75, 2.2, 212.85),
(1006, '23-Mar-2018', 22, 104, 501, 96.75, 4.2, 406.35),
(1007, '23-Mar-2018', 25, 113, 508, 48.10, 3.8, 182.78),
(1008, '24-Mar-2018', 18, 103, 500, 35.75, 0.9, 32.18),
(1009, '24-Mar-2018', 15, 115, 501, 96.75, 5.6, 541.80),
(1010, '24-Mar-2018', 15, 117, 509, 34.55, 2.4, 82.92),
(1011, '24-Mar-2018', 25, 105, 502, 125.00, 4.3, 537.50),
(1012, '24-Mar-2018', 18, 108, 501, 96.75, 3.4, 328.95),
(1013, '25-Mar-2018', 25, 115, 501, 96.75, 2.0, 193.50),
(1014, '25-Mar-2018', 22, 104, 501, 96.75, 2.8, 270.90),
(1015, '25-Mar-2018', 15, 103, 500, 35.75, 6.1, 218.08),
(1016, '25-Mar-2018', 22, 105, 502, 125.00, 4.7, 587.50),
(1017, '25-Mar-2018', 18, 117, 509, 34.55, 3.8, 131.29),
(1018, '26-Mar-2018', 25, 117, 509, 34.55, 2.2, 76.01),
(1019, '26-Mar-2018', 25, 104, 501, 96.75, 4.9, 474.08),
(1020, '26-Mar-2018', 15, 101, 502, 125.00, 3.1, 387.50),
(1021, '26-Mar-2018', 22, 108, 501, 96.75, 2.7, 261.23),
(1022, '26-Mar-2018', 22, 115, 501, 96.75, 4.9, 474.08),
(1023, '26-Mar-2018', 22, 105, 502, 125.00, 3.5, 437.50),
(1024, '26-Mar-2018', 15, 103, 500, 35.75, 3.3, 117.98),
(1025, '27-Mar-2018', 18, 117, 509, 34.55, 4.2, 145.11),
(1026, '27-Mar-2018', 18, 119, 501, 96.75, 4.8, 464.40),
(1027, '27-Mar-2018', 22, 120, 502, 125.00, 4.1, 512.50);
```

The last two rows of Assignment table are added with student names as shown below:

```
(1026, '27-Mar-2018', 18, 119, 501, 96.75, 4.8, 464.40),
(1027, '27-Mar-2018', 22, 120, 502, 125.00, 4.1, 512.50);
```

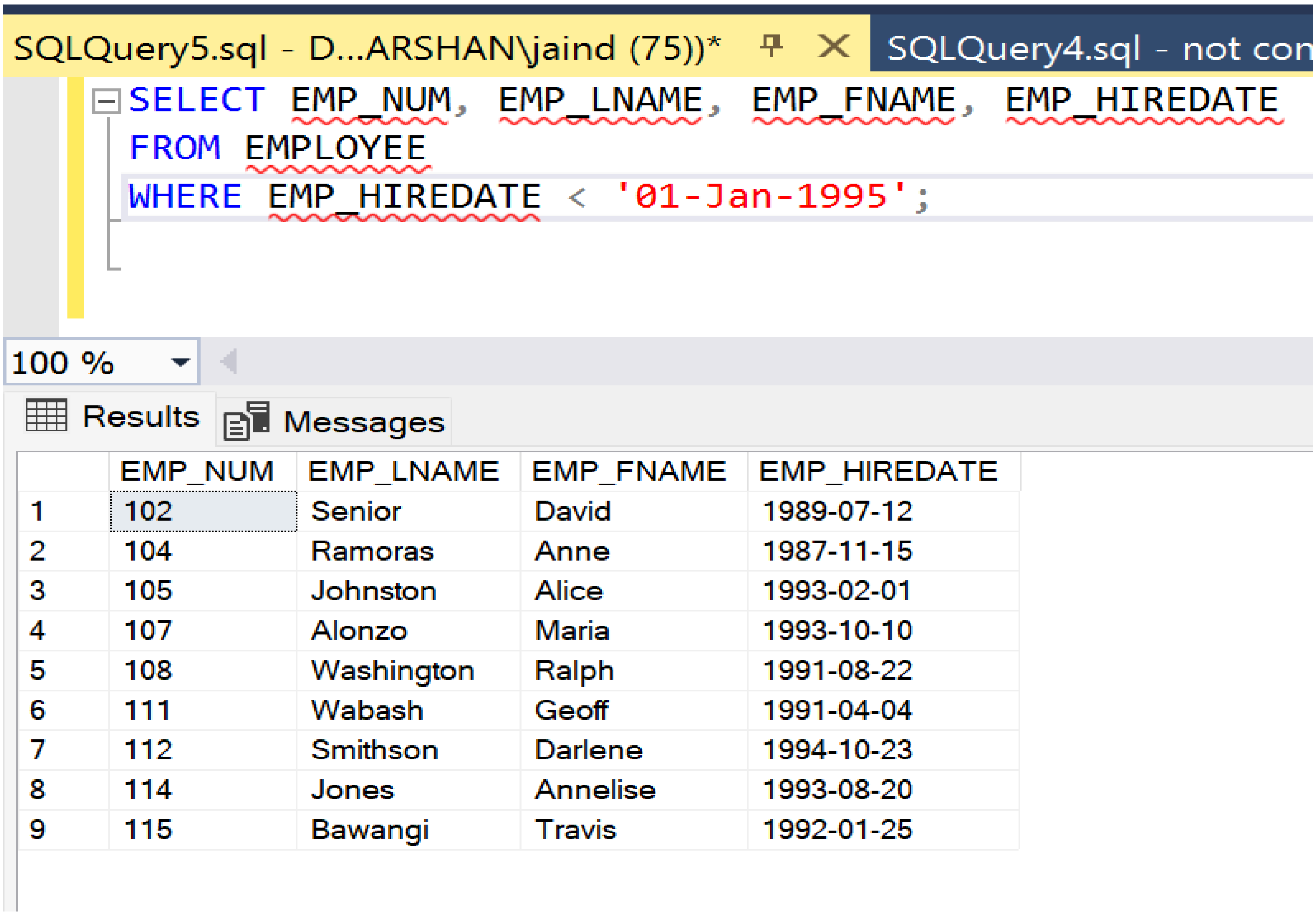

SQLQuery5.sql - D...ARSHAN\jaind (75))*
SQLQuery4.sql - not connected*
Darshan.ASS_2 - Diagram_(
100 %
Results
Messages

```
SELECT * FROM ASSIGNMENT;
```

	ASSIGN_NUM	ASSIGN_DATE	PROJ_NUM	EMP_NUM	ASSIGN_JOB	ASSIGN_CHR_HR	ASSIGN_HOURS	ASSIGN_CHARGE
1	1001	2018-03-22	18	103	500	35.75	3.5	125.13
2	1002	2018-03-22	22	117	509	34.55	4.2	145.11
3	1003	2018-03-23	18	117	509	34.55	2.0	69.10
4	1004	2018-03-23	18	103	500	35.75	5.9	210.93
5	1005	2018-03-23	25	108	501	96.75	2.2	212.85
6	1006	2018-03-23	22	104	501	96.75	4.2	406.35
7	1007	2018-03-23	25	113	508	48.10	3.8	182.78
8	1008	2018-03-24	18	103	500	35.75	0.9	32.18
9	1009	2018-03-24	15	115	501	96.75	5.6	541.80
10	1010	2018-03-24	15	117	509	34.55	2.4	82.92
11	1011	2018-03-24	25	105	502	125.00	4.3	537.50
12	1012	2018-03-24	18	108	501	96.75	3.4	328.95
13	1013	2018-03-25	25	115	501	96.75	2.0	193.50
14	1014	2018-03-25	22	104	501	96.75	2.8	270.90
15	1015	2018-03-25	15	103	500	35.75	6.1	218.08
16	1016	2018-03-25	22	105	502	125.00	4.7	587.50
17	1017	2018-03-25	18	117	509	34.55	3.8	131.29
18	1018	2018-03-26	25	117	509	34.55	2.2	76.01
19	1019	2018-03-26	25	104	501	96.75	4.9	474.08
20	1020	2018-03-26	15	101	502	125.00	3.1	387.50
21	1021	2018-03-26	22	108	501	96.75	2.7	261.23
22	1022	2018-03-26	22	115	501	96.75	4.9	474.08
23	1023	2018-03-26	22	105	502	125.00	3.5	437.50
24	1024	2018-03-26	15	103	500	35.75	3.3	117.98
25	1025	2018-03-27	18	117	509	34.55	4.2	145.11
26	1026	2018-03-27	18	119	501	96.75	4.8	464.40
27	1027	2018-03-27	22	120	502	125.00	4.1	512.50

Task 4:

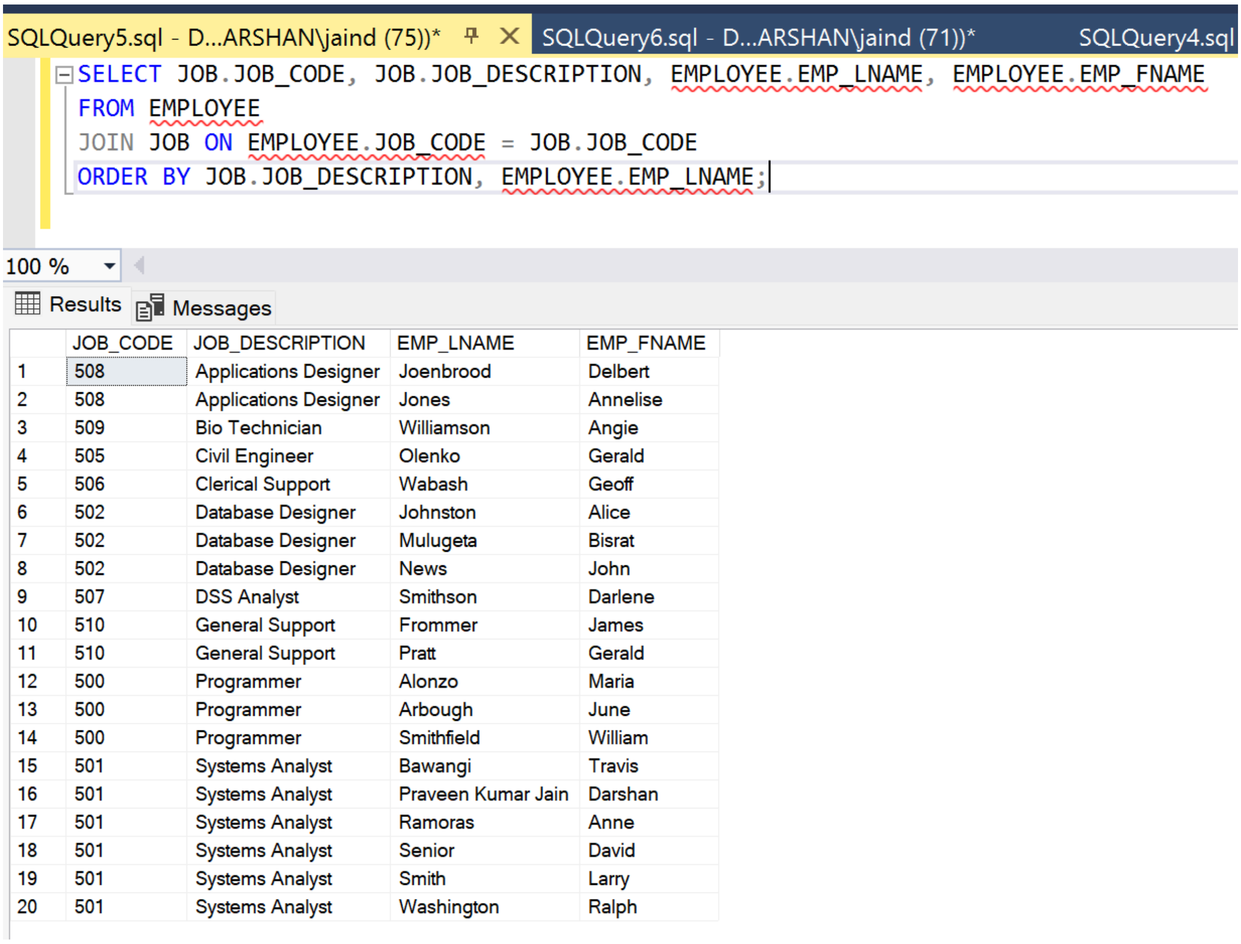
- Create and run a query to list employees who were hired before the year 1995.



```
SELECT EMP_NUM, EMP_LNAME, EMP_FNAME, EMP_HIREDATE
FROM EMPLOYEE
WHERE EMP_HIREDATE < '01-Jan-1995';
```

	EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_HIREDATE
1	102	Senior	David	1989-07-12
2	104	Ramoras	Anne	1987-11-15
3	105	Johnston	Alice	1993-02-01
4	107	Alonzo	Maria	1993-10-10
5	108	Washington	Ralph	1991-08-22
6	111	Wabash	Geoff	1991-04-04
7	112	Smithson	Darlene	1994-10-23
8	114	Jones	Annelise	1993-08-20
9	115	Bawangi	Travis	1992-01-25

- Create and run a query to list job code, job description, and employee last name, first name sorted by job description and employee last name.

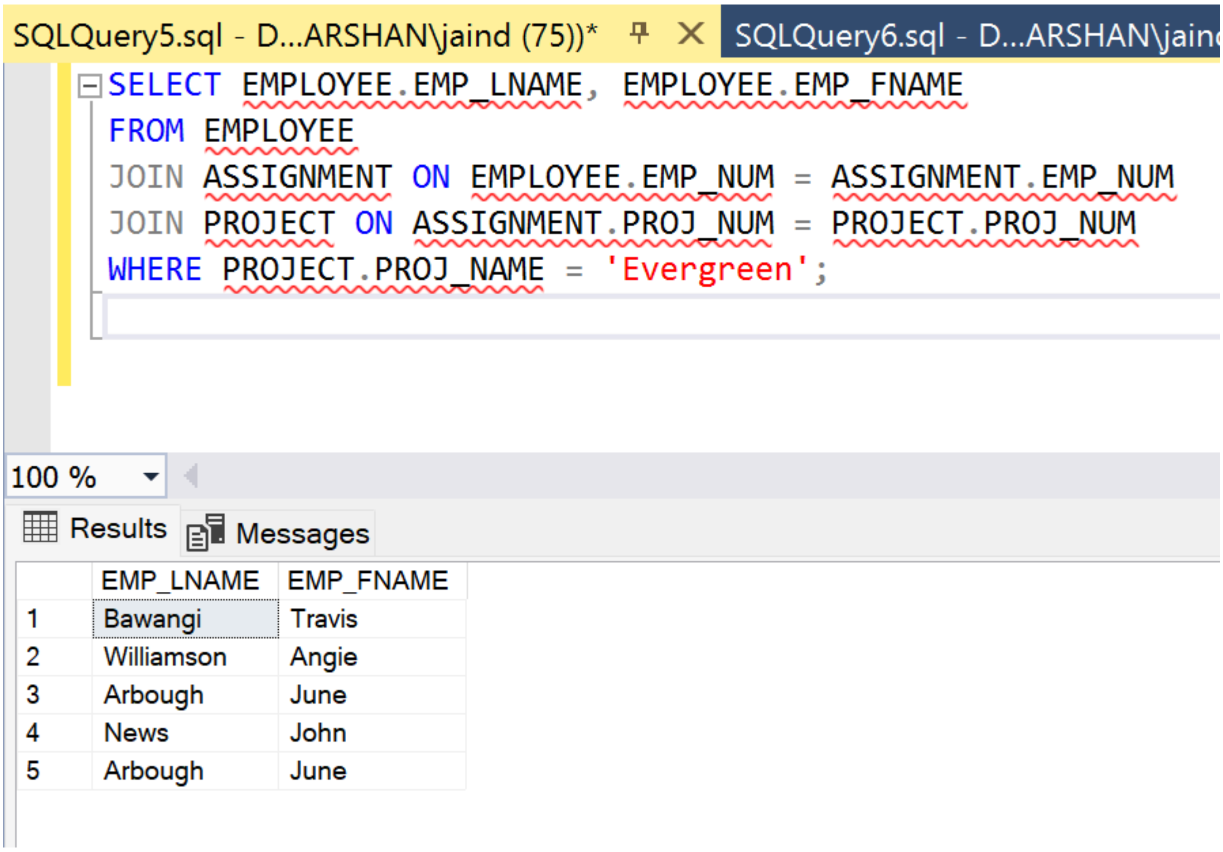


SQLQuery5.sql - D...ARSHAN\jaind (75))*
SQLQuery6.sql - D...ARSHAN\jaind (71))*
SQLQuery4.sql
100 %
Results
Messages

```
SELECT JOB.JOB_CODE, JOB.JOB_DESCRIPTION, EMPLOYEE.EMP_LNAME, EMPLOYEE.EMP_FNAME
FROM EMPLOYEE
JOIN JOB ON EMPLOYEE.JOB_CODE = JOB.JOB_CODE
ORDER BY JOB.JOB_DESCRIPTION, EMPLOYEE.EMP_LNAME;
```

	JOB_CODE	JOB_DESCRIPTION	EMP_LNAME	EMP_FNAME
1	508	Applications Designer	Joenbrood	Delbert
2	508	Applications Designer	Jones	Annelise
3	509	Bio Technician	Williamson	Angie
4	505	Civil Engineer	Olenko	Gerald
5	506	Clerical Support	Wabash	Geoff
6	502	Database Designer	Johnston	Alice
7	502	Database Designer	Mulugeta	Bisrat
8	502	Database Designer	News	John
9	507	DSS Analyst	Smithson	Darlene
10	510	General Support	Frommer	James
11	510	General Support	Pratt	Gerald
12	500	Programmer	Alonzo	Maria
13	500	Programmer	Arbough	June
14	500	Programmer	Smithfield	William
15	501	Systems Analyst	Bawangi	Travis
16	501	Systems Analyst	Praveen Kumar Jain	Darshan
17	501	Systems Analyst	Ramoras	Anne
18	501	Systems Analyst	Senior	David
19	501	Systems Analyst	Smith	Larry
20	501	Systems Analyst	Washington	Ralph

- Create and run a query to list the name of employees who were assigned to the Evergreen project.

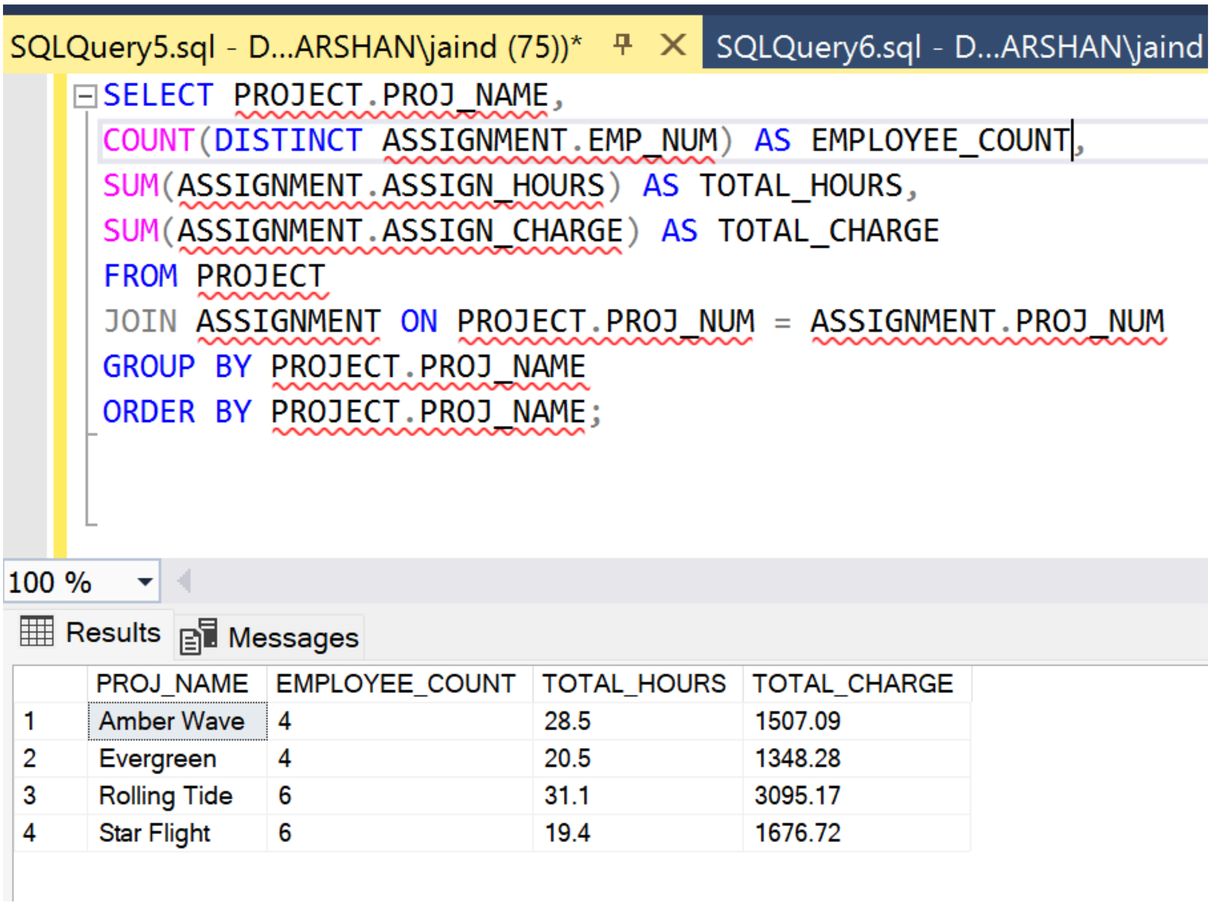


SQLQuery5.sql - D...ARSHAN\jaind (75))*
SQLQuery6.sql - D...ARSHAN\jain

```sql
SELECT EMPLOYEE.EMP_LNAME, EMPLOYEE.EMP_FNAME
FROM EMPLOYEE
JOIN ASSIGNMENT ON EMPLOYEE.EMP_NUM = ASSIGNMENT.EMP_NUM
JOIN PROJECT ON ASSIGNMENT.PROJ_NUM = PROJECT.PROJ_NUM
WHERE PROJECT.PROJ_NAME = 'Evergreen';
```

	EMP_LNAME	EMP_FNAME
1	Bawangi	Travis
2	Williamson	Angie
3	Arbough	June
4	News	John
5	Arbough	June

- Create and run a query to list the project name, the number of employees worked for the project, and the total hours and total charges assigned to each project sorted by project name.



SQLQuery5.sql - D...ARSHAN\jaind (75))*
SQLQuery6.sql - D...ARSHAN\jaind
100 %
Results
Messages

```sql
SELECT PROJECT.PROJ_NAME,
COUNT(DISTINCT ASSIGNMENT.EMP_NUM) AS EMPLOYEE_COUNT,
SUM(ASSIGNMENT.ASSIGN_HOURS) AS TOTAL_HOURS,
SUM(ASSIGNMENT.ASSIGN_CHARGE) AS TOTAL_CHARGE
FROM PROJECT
JOIN ASSIGNMENT ON PROJECT.PROJ_NUM = ASSIGNMENT.PROJ_NUM
GROUP BY PROJECT.PROJ_NAME
ORDER BY PROJECT.PROJ_NAME;
```

	PROJ_NAME	EMPLOYEE_COUNT	TOTAL_HOURS	TOTAL_CHARGE
1	Amber Wave	4	28.5	1507.09
2	Evergreen	4	20.5	1348.28
3	Rolling Tide	6	31.1	3095.17
4	Star Flight	6	19.4	1676.72

Task 5:

In task 1, we wrote commands to create the table by specifying the necessary primary keys, NOT NULL values and foreign keys. We also made sure to follow the same date type for foreign key reference to maintain data integrity.
In task 2, we have shown entity integrity, reference integrity and domain integrity in detail by specifying all the primary and foreign keys.
In task 3, we inserted sample data in all the tables with also specific rows of personalised data with our student names and executed the program. Two extra insert statements have been added to the EMPLOYEE table with student names to distinguish the table from other students and similarly two extra insert statements have been added to the ASSIGNMENT table referring to the same data added in the employee table.
In task 4, we have designed queries based on the given questions to achieve the required output.