

CPSC 5021 In-Class Exercise (SQL) Solution

The database below stores data for a consulting company that tracks all charges to projects. The charges are based on the hours each employee works on each project. The structure and contents of the database are shown below.

Table name: EMPLOYEE

| <u>EMP_NUM</u> | EMP_LNAME | EMP_FNAME | EMP_INITIAL | EMP_HIREDATE | JOB_CODE | EMP_YEAR |
|----------------|------------|-----------|-------------|--------------|----------|----------|
| 101 | News | John | G | 2000-11-08 | 502 | 12 |
| 102 | Senior | David | H | 1989-07-12 | 501 | 23 |
| 103 | Arbough | June | E | 1996-12-01 | 500 | 16 |
| 104 | Ramoras | Anne | K | 1987-11-15 | 501 | 25 |
| 105 | Johnson | Alice | K | 1993-02-01 | 502 | 19 |
| 106 | Smithfield | William | S | 2004-06-22 | 500 | 8 |
| 107 | Alonzo | Maria | D | 1993-10-10 | 501 | 8 |
| 108 | Washington | Ralph | B | 1991-08-22 | 501 | 21 |

(Note: In table “EMPLOYEE”, JOB_CODE is the foreign key that references JOB_CODE in table “JOB”.)

Table name: JOB

| <u>JOB_CODE</u> | JOB_DESCRIPTION | JOB_CHG_HOUR | JOB_LAST_UPDATE |
|-----------------|---------------------|--------------|-----------------|
| 500 | Programmer | 35.75 | 2009-11-20 |
| 501 | System Analyst | 96.75 | 2009-11-20 |
| 502 | Database Designer | 125.00 | 2010-03-24 |
| 503 | Electrical Engineer | 84.50 | 2009-11-20 |

Table name: ASSIGNMENT

| <u>ASSIGN_NUM</u> | ASSIGN_DATE | PROJ_NUM | EMP_NUM | ASSIGN_JOB | ASSIGN_CHG_HR | ASSIGN_HOURS |
|-------------------|-------------|----------|---------|------------|---------------|--------------|
| 1001 | 2012-03-22 | 18 | 103 | 500 | 84.50 | 3.5 |
| 1002 | 2012-03-22 | 18 | 102 | 501 | 84.50 | 5.9 |
| 1003 | 2012-03-22 | 25 | 108 | 501 | 96.75 | 2.2 |
| 1004 | 2012-03-22 | 22 | 102 | 501 | 96.75 | 4.2 |
| 1005 | 2012-03-22 | 18 | 103 | 500 | 84.50 | 0.9 |
| 1006 | 2012-03-23 | 25 | 107 | 501 | 105.00 | 4.3 |
| 1007 | 2012-03-23 | 18 | 108 | 501 | 96.75 | 3.4 |
| 1008 | 2012-03-23 | 22 | 104 | 501 | 96.75 | 2.8 |
| 1009 | 2012-03-23 | 15 | 103 | 500 | 84.50 | 6.1 |
| 1010 | 2012-03-23 | 22 | 105 | 502 | 105.00 | 4.7 |
| 1011 | 2012-03-24 | 25 | 106 | 500 | 110.50 | 4.9 |
| 1012 | 2012-03-24 | 15 | 101 | 502 | 125.00 | 3.1 |
| 1013 | 2012-03-24 | 22 | 108 | 501 | 110.50 | 2.7 |
| 1014 | 2012-03-24 | 22 | 105 | 502 | 125.00 | 3.5 |

(Note: In table “ASSIGNMENT”, the foreign key is EMP_NUM, ASSIGN_JOB and PROJ_NUM. EMP_NUM references EMP_NUM in table “EMPLOYEE”; ASSIGN_JOB references JOB_CODE in table “JOB”; PROJ_NUM references PROJ_NUM in table “PROJECT”).

Table name: PROJECT

| PROJ_NUM | PROJ_NAME | PROJ_VALUE | PROJ_BALANCE | EMP_NUM |
|----------|--------------|------------|--------------|---------|
| 15 | Evergreen | 1453500.00 | 1002350.00 | 103 |
| 18 | Amber Wave | 3500500.00 | 2110346.00 | 108 |
| 22 | Rolling Tide | 805000.00 | 500345.20 | 102 |
| 25 | Starlight | 2850500.00 | 2309880.00 | 107 |

(Note: EMP_NUM in this table records the ID for a project’s manager. It is a foreign key which references EMP_NUM in table “EMPLOYEE”)

Given the structure and contents of the database, use SQL commands to answer questions below.

(1) Write the SQL code that will create the table structures for the table EMPLOYEE, JOB, ASSIGNMENT, and PROJECT, separately. The table structures are summarized below.

EMPLOYEE

| ATTRIBUTE NAME | DATA TYPE |
|----------------|-------------|
| EMP_NUM | CHAR(3) |
| EMP_LNAME | VARCHAR(15) |
| EMP_FNAME | VARCHAR(15) |
| EMP_INITIAL | CHAR(1) |
| EMP_HIREDATE | DATE |
| JOB_CODE | CHAR(3) |
| EMP_YEAR | SMALLINT |

JOB

| ATTRIBUTE NAME | DATA TYPE |
|-----------------|---------------|
| JOB_CODE | CHAR(3) |
| JOB_DESCRIPTION | VARCHAR(50) |
| JOB_CHG_HOUR | DECIMAL(5, 2) |
| JOB_LAST_UPDATE | DATE |

ASSIGNMENT

| ATTRIBUTE NAME | DATA TYPE |
|----------------|---------------|
| ASSIGN_NUM | CHAR(4) |
| ASSIGN_DATE | DATE |
| PROJ_NUM | CHAR(2) |
| EMP_NUM | CHAR(3) |
| ASSIGN_JOB | CHAR(3) |
| ASSIGN_CHG_HR | DECIMAL(5, 2) |
| ASSIGN_HOURS | DECIMAL(3, 1) |

PROJECT

| ATTRIBUTE NAME | DATA TYPE |
|----------------|----------------|
| PROJ_NUM | CHAR(2) |
| PROJ_NAME | VARCHAR(50) |
| PROJ_VALUE | DECIMAL(12, 2) |
| PROJ_BALANCE | DECIMAL(12, 2) |
| EMP_NUM | CHAR(3) |

Sol:

```
create table job (  
  job_code char(3),  
  job_description varchar(50),  
  job_chg_hour decimal(5,2),  
  job_last_update date,  
  primary key(job_code)  
);  
  
create table employee (  
  emp_num char(3) primary key,  
  emp_lname varchar(15) not null,  
  emp_fname varchar(15) not null,  
  emp_initial char(1),  
  emp_hiredate date,  
  job_code char(3),  
  emp_year smallint,  
  foreign key (job_code) references job(job_code)  
);
```

```
create table project (  
  proj_num char(2) not null,  
  proj_name varchar(50) not null,  
  proj_value decimal(12,2) not null,  
  proj_balance decimal(12,2) not null,  
  emp_num char(3) not null,  
  primary key(proj_num),  
  foreign key (emp_num) references employee(emp_num)  
);  
  
create table assignment (  
  assign_num char(4) not null,  
  assign_date date not null,  
  proj_num char(2) not null,  
  emp_num char(3) not null,  
  assign_job char(3) not null,  
  assign_chg_hr decimal(5,2) not null,  
  assign_hours decimal(3,1) not null,  
  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)  
);
```

(2) Write the SQL code to enter data in each table.

Sol:

```
insert into job values ('500', 'Programmer', 35.75, '2009-11-20');

insert into job values ('501', 'System Analyst', 96.75, '2009-11-20'),
('502', 'Database Designer', 125.00, '2010-03-24'),
('503', 'Electrical Engineer', 84.50, '2009-11-20');

insert into employee values ('101', 'News', 'John', 'G', '2000-11-08', '502', 12);
insert into employee values ('102', 'Senior', 'David', 'H', '1989-07-12', '501', 23),
('103', 'Arbough', 'June', 'E', '1996-12-01', '500', 16),
('104', 'Ramoras', 'Anne', 'K', '1987-11-15', '501', 25),
('105', 'Johnson', 'Alice', 'K', '1993-02-01', '502', 19),
('106', 'Smithfield', 'William', 'S', '2004-06-22', '500', 8),
('107', 'Alonzo', 'Maria', 'D', '1993-10-10', '501', 8),
('108', 'Washington', 'Ralph', 'B', '1991-08-22', '501', 21);

insert into project 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', 'Starlight', 2850500.00, 2309880.00, '107');

insert into assignment values ('1001', '2012-03-22', '18', '103', '500', 84.50, 3.5),
('1002', '2012-03-22', '18', '102', '501', 84.50, 5.9),
('1003', '2012-03-22', '25', '108', '501', 96.75, 2.2),
('1004', '2012-03-22', '22', '102', '501', 96.75, 4.2),
('1005', '2012-03-22', '18', '103', '500', 84.50, 0.9),
('1006', '2012-03-23', '25', '107', '501', 105.00, 4.3);
(the subsequent rows are skipped here)
```

(3) Write the SQL code that will list all attributes in the EMPLOYEE table for a job code of 502.

Sol:

```
SELECT *
FROM employee
WHERE JOB_CODE = '502';
```

(4) Write the SQL code that will list values of “PROJ_NUM” and “PROJ_NAME” of the PROJECT table.

```
select proj_num, proj_name
from project;
```

(5) Write the SQL code that will list all attributes in the ASSIGNMENT table with ASSIGN_HOURS > 3

sol:

```
select * from assignment
where assign_hours > 3;
```

(6) Write the SQL code that will list ASSIGN_NUM, ASSIGN_DATE, ASSIGN_CHG_HR*ASSIGN_HOURS (*note: name this product as assign_charge*) in the table ASSIGNMENT with ASSIGN_DATE later than 2012-03-21.

sol:

```
select assign_num, assign_date, assign_chg_hr * assign_hours as assign_charge from assignment
where assign_date > '2012-03-21';
```

(7) Write the SQL code that will list all attributes in the table “PROJECT” with PROJ_NUM = 22 or PROJ_NUM = 25.

Sol:

```
select * from project
where proj_num = 22
or proj_num = 25;
```

(8) Write the SQL code required to list all employees whose last names start with Smith.

Sol:

```
select * from employee
where emp_lname like 'Smith%';
```

(9) Write the SQL code to select from the table “JOB” all jobs which job_code value appear in the EMPLOYEE table.

Sol:

```
select * from job
where job_code in
(select job_code from employee);
```

(10) Write the SQL code that will produce a listing for the data in the EMPLOYEE table in descending order by EMP_YEAR.

Sol:

```
select * from employee
order by emp_year desc;
```

(11) Write the SQL code that will list only the distinct EMP_NUM in the table “ASSIGNMENT”.

Sol:

```
select distinct emp_num from assignment;
```

(12) Write the SQL code to find the average PROJ_VALUE in the table “PROJECT”.

Sol:

```
select avg(proj_value) from project;
```

(13) Write the SQL code to count the number of distinct EMP_NUM in the table “ASSIGNMENT”.

Sol:

```
select count(distinct emp_num) from assignment;
```

(14) Write the SQL code to list all attributes of the project that has the largest amount of PROJ_VALUE.

Sol:

```
select * from project  
where proj_value = (select max(proj_value) from project);
```

(15) Write the SQL code to list all attributes of the project(s) which PROJ_VALUE is higher than the average PROJ_VALUE. Sort the result by PROJ_BALANCE in ascending order.

Sol:

```
select * from project  
where proj_value >  
(select avg(proj_value) from project)  
order by proj_balance asc;
```

(16) Write the SQL code to find the numbers of employees that each project has been assigned to.

sol:

```
SELECT proj_num, COUNT( DISTINCT emp_num)  
FROM assignment  
GROUP BY proj_num;
```

(17) Write the SQL code to list the EMP_NUM and the number of projects s/he has been assigned to for employees who has been assigned to at least 2 projects.

Sol:

```
SELECT emp_num, COUNT( DISTINCT proj_num) as workload  
FROM assignment  
GROUP BY emp_num  
HAVING (workload >= 2)
```

(18) Write the SQL code to change the job code to 501 for the person whose employee number (EMP_NUM) is 107.

Sol:

```
UPDATE employee
SET JOB_CODE = '501'
WHERE EMP_NUM = '107'
```

(19) Write the SQL code to delete the row for William Smithfield, who was hired on June 22, 2004, and whose job code is 500.

Sol:

```
DELETE FROM employee
WHERE EMP_LNAME = 'Smithfield'
AND EMP_FNAME = 'William'
AND EMP_HIREDATE = '2004-06-22'
AND JOB_CODE = '500';
```

(20) Write the SQL code to create a copy of EMPLOYEE, naming the copy EMP_1. Then write the SQL code that will add the attributes EMP_PCT and PROJ_NUM to its structure. The EMP_PCT is the bonus percentage to be paid to each employee. The new attribute characteristics are:

```
EMP_PCT DECIMAL(4, 2)
PROJ_NUM CHAR(3)
```

Solution A:

```
CREATE TABLE EMP_1 (
EMP_NUM CHAR(3) NOT NULL UNIQUE,
EMP_LNAME VARCHAR(15) NOT NULL,
EMP_FNAME VARCHAR(15) NOT NULL,
EMP_INITIAL CHAR(1),
EMP_HIREDATE DATE NOT NULL,
JOB_CODE CHAR(3) NOT NULL,
EMP_YEAR SMALLINT,
PRIMARY KEY (EMP_NUM),
FOREIGN KEY (JOB_CODE) REFERENCES JOB (JOB_CODE));
```

```
INSERT INTO EMP_1 SELECT * FROM EMPLOYEE;
```

```
ALTER TABLE EMP_1
ADD (EMP_PCT DECIMAL(4,2)),
ADD (PROJ_NUM CHAR(3));
```

Solution B:

```
CREATE TABLE EMP_1 AS SELECT * FROM employee;
```

```
ALTER TABLE EMP_1
ADD (EMP_PCT DECIMAL (4,2)),
ADD (PROJ_NUM CHAR(3)),
ADD PRIMARY KEY (EMP_NUM),
ADD FOREIGN KEY (JOB_CODE) REFERENCES JOB(JOB_CODE);
```

(21) Write the SQL code that will change the EMP_YEAR to 14 for employees who were hired before January 1, 1994, and whose job code is at least 501.

```
UPDATE employee  
SET emp_year = 14  
WHERE EMP_HIREDATE <= '1994-01-01'  
AND JOB_CODE >= '501';
```