In [1]:
```
#Install pysqlite3 for python and import pandas to use later
#!pip install pysqlite3
from sqlite3 import dbapi2 as sqlite3
print(sqlite3.sqlite_version)
import pandas as pd
from IPython.display import display, HTML
```

```
3.45.3
```

Specify the following queries in SQL on the COMPANY relational database schema shown in Figure 5.5. Show the result of each query if it is applied to the COMPANY database in Figure 5.6. a. Retrieve the names of all employees in department 5 who work more than 10 hours per week on the ProductX project. b. List the names of all employees who have a dependent with the same first name as themselves. c. Find the names of all employees who are directly supervised by 'Franklin Wong'.

In [2]:
```
dbname = "Company.db"

def printSqlResults(cursor, tblName):
  try:
    df = pd.DataFrame(cursor.fetchall(), columns=[i[0] for i in cursor.description])
    display(HTML("<b><font color=Green> " + tblName + "</font></b>" + df.to_html(inde
  except:
    pass

def runSql(caption, query):
  conn = sqlite3.connect(dbname) # Connect to the database
  cursor = conn.cursor() # Create a cursor (think: it's like a "pointer")
  cursor.execute(query) # Execute the query
  printSqlResults(cursor, caption) # Print the results
  conn.close()

def runSqlWithCommit(caption, query):
  conn = sqlite3.connect(dbname) # Connect to the database
  cursor = conn.cursor() # Create a cursor (think: it's like a "pointer")
  cursor.execute(query) # Execute the query
  printSqlResults(cursor, caption) # Print the results
  conn.commit()
  conn.close()


def runStepByStepSql(query, fromline):
  lines = query.strip().split('\n')
  for lineidx in range(fromline, len(lines)):
    partial_query = '\n'.join(lines[:lineidx])
    caption = 'Query till line:' +  partial_query
    runSql(caption, partial_query + ';')
```

In [3]:
```
conn = sqlite3.connect(dbname)
cursor = conn.cursor()

#Create and Insert Data in EMPLOYEE Table
cursor.execute("""
CREATE TABLE EMPLOYEE(
   Fname     VARCHAR(8) NOT NULL
  ,Minit     VARCHAR(1) NOT NULL
  ,Lname     VARCHAR(7) NOT NULL
  ,Ssn       INTEGER  NOT NULL PRIMARY KEY
  ,Bdate     DATE  NOT NULL
  ,Address   VARCHAR(24) NOT NULL
  ,Sex       VARCHAR(1) NOT NULL
  ,Salary    INTEGER  NOT NULL
  ,Super_ssn INTEGER
  ,Dno       INTEGER  NOT NULL
);
""")

cursor.execute("""
INSERT INTO EMPLOYEE(Fname,Minit,Lname,Ssn,Bdate,Address,Sex,Salary,Super_ssn,Dno) VA
('Franklin','T','Wong',333445555,'1955-12-08','638 Voss, Houston, TX','M',40000,88866
('Jennifer','J','Wallace',999887777,'1968-01-19','3321 Castle, Spring, TX','F',43000,
('Alicia','S','Zelaya',987654321,'1941-06-20','291 Berry, Bellaire, TX','F',25000,888
('Ramesh','K','Narayan',666884444,'1962-09-15','975 Fire Oak, Humble, TX','M',38000,3
('Joyce','A','English',453453453,'1972-07-31','5631 Rice, Houston, TX','F',25000,3334
('Ahmad','V','Jabbar',987987987,'1969-03-29','980 Dallas, Houston, TX','M',25000,9876
('James','E','Borg',888665555,'1937-11-10','450 Stone, Houston, TX','M',55000,NULL,1)
""")

#Create and Insert Data in DEPARTMENT Table
cursor.execute("""
CREATE TABLE DEPARTMENT(
   Department VARCHAR(14) NOT NULL
  ,Dnumber     VARCHAR(7) NOT NULL PRIMARY KEY
  ,Mgr_ssn      VARCHAR(9) NOT NULL
  ,Mgr_start_date     VARCHAR(14) NOT NULL
);
""")

cursor.execute("""
INSERT INTO DEPARTMENT(Department,Dnumber,Mgr_ssn,Mgr_start_date) VALUES
('Research','5','333445555','1988-05-22'),
('Administration','4','987654321','1995-01-01'),
('Headquarters','1','888665555','1981-06-19');
""")

#Create and Insert Data in DEPT_LOCATION Table
cursor.execute("""
CREATE TABLE DEPT_LOCATION(
   Dnumber   INTEGER  NOT NULL
  ,Dlocation VARCHAR(9) NOT NULL
  ,PRIMARY KEY(Dnumber,Dlocation)
);
""")

cursor.execute("""
INSERT INTO DEPT_LOCATION(Dnumber,Dlocation) VALUES (1,'Houston'),
(4,'Stafford'),
(5,'Bellaire'),
(5,'Sugarland'),
(5,'Houston');
""")

#Create and Insert Data in PROJECT Table
cursor.execute("""
CREATE TABLE PROJECT(
   Pname     VARCHAR(15) NOT NULL
  ,Pnumber   INTEGER  NOT NULL PRIMARY KEY
  ,Plocation VARCHAR(9) NOT NULL
  ,Dnum      INTEGER  NOT NULL
);
""")

cursor.execute("""
INSERT INTO PROJECT(Pname,Pnumber,Plocation,Dnum) VALUES ('ProductX',1,'Bellaire',5),
('ProductY',2,'Sugarland',5),
('ProductZ',3,'Houston',5),
('Computerization',10,'Stafford',4),
('Reorganization',20,'Houston',1),
('Newbenefits',30,'Stafford',4);
""")

#Create and Insert Data in WORKS_ON Table
cursor.execute("""
CREATE TABLE WORKS_ON(
   Essn  INTEGER  NOT NULL
  ,Pno   INTEGER  NOT NULL
  ,Hours NUMERIC(4,1)
  ,PRIMARY KEY(Essn,Pno)
);
""")

cursor.execute("""
INSERT INTO WORKS_ON(Essn,Pno,Hours) VALUES (123456789,1,32.5),
(123456789,2,7.5),
(666884444,3,40),
(453453453,1,20),
(453453453,2,20),
(333445555,2,10),
(333445555,3,10),
(333445555,10,10),
(333445555,20,10),
(999887777,30,30),
(999887777,10,10),
(987987987,10,35),
(987987987,30,5),
(987654321,30,20),
(987654321,20,15),
(888665555,20,NULL);
""")

#Create and Insert Data in Dependent Table
cursor.execute("""
CREATE TABLE DEPENDENT(
   Essn           INTEGER  NOT NULL
  ,Dependent_name VARCHAR(9) NOT NULL
  ,Sex            VARCHAR(1) NOT NULL
  ,Bdate          DATE  NOT NULL
  ,Relationship   VARCHAR(8) NOT NULL
  ,PRIMARY KEY(Essn,Dependent_name)
);
""")

cursor.execute("""
INSERT INTO DEPENDENT(Essn,Dependent_name,Sex,Bdate,Relationship) VALUES (333445555,'
(333445555,'Theodore','M','1983-10-25','Son'),
(333445555,'Joy','F','1958-05-03','Spouse'),
(987654321,'Abner','M','1942-02-28','Spouse'),
(123456789,'Michael','M','1988-01-04','Son'),
(123456789,'Alice','F','1988-12-30','Daughter'),
(123456789,'Elizabeth','F','1967-05-05','Spouse');
""")


conn.commit()
conn.close()
```

In [4]:

```
runSql('EMPLOYEE', "select * from EMPLOYEE;")
runSql('DEPARTMENT', "select * from DEPARTMENT;")
runSql('DEPT_Location', "select * from DEPT_Location;")
runSql('PROJECT', "select * from PROJECT;")
runSql('WORKS_ON', "select * from WORKS_ON;")
runSql('DEPENDENT', "select * from DEPENDENT;")
```

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555.0	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555.0	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555.0	5
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555.0	5
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NaN	1
Alicia	S	Zelaya	987654321	1941-06-20	291 Berry, Bellaire, TX	F	25000	888665555.0	4
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321.0	4
Jennifer	J	Wallace	999887777	1968-01-19	3321 Castle, Spring, TX	F	43000	987654321.0	4

DEPARTMENT

Department	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_Location

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

WORKS_ON

	Essn	Pno	Hours
	123456789	1	32.5
	123456789	2	7.5
	666884444	3	40.0
	453453453	1	20.0
	453453453	2	20.0
	333445555	2	10.0
	333445555	3	10.0
	333445555	10	10.0
	333445555	20	10.0
	999887777	30	30.0
	999887777	10	10.0
	987987987	10	35.0
	987987987	30	5.0
	987654321	30	20.0
	987654321	20	15.0
	888665555	20	NaN

DEPENDENT

	Essn	Dependent_name	Sex	Bdate	Relationship
	333445555	Alice	F	1986-04-05	Daughter
	333445555	Theodore	M	1983-10-25	Son
	333445555	Joy	F	1958-05-03	Spouse
	987654321	Abner	M	1942-02-28	Spouse
	123456789	Michael	M	1988-01-04	Son
	123456789	Alice	F	1988-12-30	Daughter
	123456789	Elizabeth	F	1967-05-05	Spouse

Problem 1

Specify the following queries in SQL on the COMPANY relational database schema. Show the result of each query if it is applied to the COMPANY database.

- Retrieve the names of all employees in department 5 who work more than 10 hours per week on the ProductX project.
- List the names of all employees who have a dependent with the same first name as themselves.
- Find the names of all employees who are directly supervised by ‘Franklin Wong’.

In [5]:
```
print("\n")
emp_projX = """
SELECT CONCAT(Fname, ' ', Lname) as Employee FROM
DEPARTMENT dept
JOIN EMPLOYEE emp ON dept.Dnumber=emp.Dno
JOIN WORKS_ON work_on ON work_on.Essn=emp.ssn
JOIN PROJECT project ON project.Pnumber=work_on.Pno
WHERE project.Pname = 'ProductX' and dept.Dnumber = 5 and work_on.Hours>10;
"""
runSql('Employees from ProductX', emp_projX)
print("\n")
```

Employees from ProductX

Employee
John Smith
Joyce English

In [6]:
```
print("\n")
dependentSame = """
SELECT * FROM
DEPENDENT depn
JOIN EMPLOYEE emp ON depn.Essn=emp.ssn
WHERE depn.Dependent_name=emp.Fname;
"""
runSql('Employees with dependent name same as First Name', dependentSame)
print("\n")
```

Employees with dependent name same as First Name

Essn	Dependent_name	Sex	Bdate	Relationship	Fname	Minit	Lname	Ssn	Bdate	Address	Se

In [7]:
```
print("\n")
supervisorFW = """
SELECT CONCAT(Fname, ' ', Lname) as Employee FROM EMPLOYEE WHERE Super_ssn IN
(SELECT Ssn FROM EMPLOYEE WHERE Fname = 'Franklin' and Lname = 'Wong');
"""
runSql('Employees supervised by Franklin Wong', supervisorFW)
print("\n")
```

Employees supervised by Franklin Wong

Employee
John Smith
Joyce English
Ramesh Narayan

Problem 2

Specify the following query on the database in Figure 5.5 in SQL. Show the query results if the query is applied to the database state in Figure 5.6.

- For each project whose average employee salary is more than $27,000, retrieve the project name and the number of employees working on that project.

In [8]:
```
avgEmp = """
SELECT project.Pname, COUNT(emp.Ssn) FROM EMPLOYEE emp
JOIN WORKS_ON works_on ON emp.Ssn=works_on.Essn
JOIN PROJECT project ON project.Pnumber=works_on.Pno
GROUP BY project.Pname HAVING AVG(Salary) > 27000;
"""
runSql('Employees whose average employee salary is more than $27,000', avgEmp)
```

Employees whose average employee salary is more than $27,000

Pname	COUNT(emp.Ssn)
Computerization	3
Newbenefits	3
ProductX	2
ProductY	3
ProductZ	2
Reorganization	3

Problem 3

In SQL, show the following queries on the database in Figure 5.5 using the concept of nested queries and other concepts described in chapter 7. Additionally, list the results of these queries.

- Retrieve the names of all employees who work in the department that has the employee with the highest salary among all employees.
- Retrieve the names of all employees whose supervisor's supervisor has '123456789' for Ssn.
- Retrieve the names of employees who make at least $10,000 more than the employee who is paid the least in the company.

In [9]:
```
print("\n")

highestSal = """
SELECT CONCAT(Fname, ' ', Lname) as Name FROM EMPLOYEE WHERE Dno = (SELECT Dno FROM E
"""
runSql('Names of all employees who work in the department that has the employee with

print("\n")

Super_ssn = """
SELECT * FROM EMPLOYEE WHERE Super_ssn = (SELECT Ssn FROM EMPLOYEE WHERE Super_ssn =
"""
runSql('Names of all employees whose supervisor's supervisor has '123456789' for Ssn'

print("\n")

Least_ssn = """
SELECT CONCAT(Fname, ' ', Lname) FROM EMPLOYEE WHERE Salary >= (SELECT Salary + 10000
"""
runSql('Names of employees who make at least $10,000 more than the employee who is pa

print("\n")
```

Names of all employees who work in the department that has the employee with the highest salary among all employees

Name
James Borg

Names of all employees whose supervisor's supervisor has '123456789' for Ssn

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

Names of employees who make at least $10,000 more than the employee who is paid the least in the company.

CONCAT(Fname, ' ', Lname)
Franklin Wong
Ramesh Narayan
James Borg
Jennifer Wallace

In [10]:
```
conn = sqlite3.connect(dbname)
cursor = conn.cursor()

#Create and Insert Data in STUDENT Table
cursor.execute("""
CREATE TABLE STUDENT(
   Name           VARCHAR(5) NOT NULL
  ,Student_number INTEGER  NOT NULL
  ,Class          INTEGER  NOT NULL
  ,Major          VARCHAR(2) NOT NULL
);
""")

cursor.execute("""
INSERT INTO STUDENT(Name,Student_number,Class,Major) VALUES ('Smith',17,1,'CS'),
('Brown',8,2,'CS');
""")

#Create and Insert Data in COURSE Table
cursor.execute("""
CREATE TABLE COURSE(
   Name           VARCHAR(5) NOT NULL
  ,Course_number INTEGER  NOT NULL
  ,Class          INTEGER  NOT NULL
  ,Major          VARCHAR(2) NOT NULL
);
""")

cursor.execute("""
INSERT INTO COURSE(Name,Course_number,Class,Major) VALUES ('Intro to Computer Science
('Data Structures','CS3320',4,'CS'),
('Discrete Mathematics','MATH2410',3,'MATH'),
('Database','CS3380',3,'CS');

""")

#Create and Insert Data in SECTION Table
cursor.execute("""
CREATE TABLE SECTION(
   Section_identifier INTEGER  NOT NULL
  ,Course_number      VARCHAR(8) NOT NULL
  ,Semester           VARCHAR(6) NOT NULL
  ,Year               INTEGER  NOT NULL
  ,Instructor         VARCHAR(8) NOT NULL
);
""")

cursor.execute("""
INSERT INTO SECTION(Section_identifier,Course_number,Semester,Year,Instructor) VALUES
(92,'CS1310','Fall',07,'Anderson'),
(102,'CS3320','Spring',08,'Knuth'),
(112,'MATH2410','Fall',08,'Chang'),
(119,'CS1310','Fall',08,'Anderson'),
(135,'CS3380','Fall',08,'Stone');
""")

#Create and Insert Data in GRADE_REPORT Table
cursor.execute("""
CREATE TABLE GRADE_REPORT(
   Student_number     INTEGER  NOT NULL
  ,Section_identifier INTEGER  NOT NULL
  ,Grade              VARCHAR(1) NOT NULL
);
""")

cursor.execute("""
INSERT INTO GRADE_REPORT(Student_number,Section_identifier,Grade) VALUES (17,112,'B')
(17,119,'C'),
(8,85,'A'),
(8,92,'A'),
(8,102,'B'),
(8,135,'A');
""")

#Create and Insert Data in PREREQUISITE Table
cursor.execute("""
CREATE TABLE PREREQUISITE(
   Course_number       VARCHAR(6) NOT NULL
  ,Prerequisite_number VARCHAR(8) NOT NULL
);
""")

cursor.execute("""
INSERT INTO PREREQUISITE(Course_number,Prerequisite_number) VALUES ('CS3380','CS3320'
('CS3380','MATH2410'),
('CS3320','CS1310');
""")

conn.commit()
conn.close()
```

In [11]:

```
runSql('STUDENT', "select * from STUDENT;")
runSql('COURSE', "select * from COURSE;")
runSql('SECTION', "select * from SECTION;")
runSql('GRADE_REPORT', "select * from GRADE_REPORT;")
runSql('PREREQUISITE', "select * from PREREQUISITE;")
```

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Name	Course_number	Class	Major
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	7	King
92	CS1310	Fall	7	Anderson
102	CS3320	Spring	8	Knuth
112	MATH2410	Fall	8	Chang
119	CS1310	Fall	8	Anderson
135	CS3380	Fall	8	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Problem 4

Specify the following queries in SQL on the database schema in Figure 1.2.

- Retrieve the number of all straight-A students (students who have a grade of A in all their courses).
- Retrieve the names and major departments of all students who do not have a grade of A in any of their courses

In [17]:
```
straight_A = """
SELECT COUNT(DISTINCT Student_number)
FROM ( SELECT Student_number FROM GRADE_REPORT EXCEPT SELECT Student_number FROM GRAD
"""
runSql('Number of all straight-A students', straight_A)

print("\n")
```

```
nstraight_A = """
SELECT DISTINCT s.Name, s.Major FROM STUDENT s WHERE s.Student_number IN
( SELECT gr.Student_number FROM GRADE_REPORT gr EXCEPT SELECT gr.Student_number
FROM GRADE_REPORT gr WHERE gr.Grade = 'A');
"""
runSql('Number of non all straight-A students', nstraight_A)
```

Number of all straight-A students

COUNT(DISTINCT Student_number)
0

Number of non all straight-A students

Name	Major
Smith	CS

Problem 5

Imagine you are designing a table to store recent transactions for an online shopping platform and there are 1 trillion transactions. You want to record the following information:

- user id
- user name
- item id
- item name
- transaction id
- amount of money for the transaction (e.g. 7.81, 470.80, etc) (In dollars)

1. What data type should you use for each column? You need to fill one of the following data types: byte, short, int, long, float, double, boolean, char.

Ans: The datatypes should be:

- user id: long
- user name: char[20]
- item id: long
- item name: char[20]
- transaction id: long
- amount of money for the transaction: double

2. What is the size of each row in bytes? Think about the size of each column by selecting proper data types. You need to select the most suitable data type for each column by considering efficiency.

Ans: The size of each row would be long(4bytes) + char(20byte) + long(4bytes) + char(20byte) + long(4bytes) + double(8bytes) = 60 bytes

3. What is the size of the table in TB?

Ans: 60TB

```
In [ ]:
```