RED TEAM - Blend SQL Statements and Queries

William Borum, James D’Arpino, Stephan Karnasky, Christine Robertson

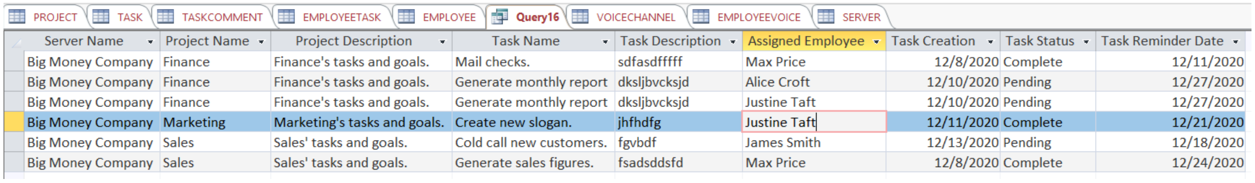
```
CREATE TABLE EMPLOYEE(
EmployeeID VARCHAR(6) NOT NULL,
Name VARCHAR(50),
Username VARCHAR(30),
Email VARCHAR(50),
Password VARCHAR(20),
Gender VARCHAR(1),
Age INT,
CONSTRAINT EMPLOYEE_PK PRIMARY KEY(EmployeeID));

CREATE TABLE SERVER(
ServerID VARCHAR(6) NOT NULL,
Name VARCHAR(50),
CreationDate DATE,
CONSTRAINT SERVER_PK PRIMARY KEY(ServerID));

CREATE TABLE VOICECHANNEL(
VoiceChannelID VARCHAR(6) NOT NULL,
ServerID VARCHAR(6) NOT NULL,
Name VARCHAR(50),
CreationDate DATE,
ChannelType VARCHAR(5),
CONSTRAINT VOICECHANNEL_PK PRIMARY KEY(VoiceChannelID),
CONSTRAINT VOICECHANNEL_FK FOREIGN KEY(ServerID) REFERENCES SERVER(ServerID));

CREATE TABLE EMPLOYEEVOICE(
EmployeeID VARCHAR(6) NOT NULL,
VoiceChannelID VARCHAR(6) NOT NULL,
Mute VARCHAR(1),
ShareScreen VARCHAR(1),
ShowVideo VARCHAR(1),
CONSTRAINT EMPLOYEEVOICE_PK PRIMARY KEY(EmployeeID,VoiceChannelID),
CONSTRAINT EMPLOYEEVOICE_FK1 FOREIGN KEY(EmployeeID) REFERENCES EMPLOYEE(EmployeeID),
CONSTRAINT EMPLOYEEVOICE_FK2 FOREIGN KEY(VoiceChannelID) REFERENCES
VOICECHANNEL(VoiceChannelID));

CREATE TABLE EMPLOYEESERVER(
EmployeeID VARCHAR(6) NOT NULL,
ServerID VARCHAR(6) NOT NULL,
JoinDate DATE,
CONSTRAINT EMPLOYEESERVER_PK PRIMARY KEY(EmployeeID,ServerID),
CONSTRAINT EMPLOYEESERVER_FK1 FOREIGN KEY(EmployeeID) REFERENCES EMPLOYEE(EmployeeID),
CONSTRAINT EMPLOYEESERVER_FK2 FOREIGN KEY(ServerID) REFERENCES SERVER(ServerID));


CREATE TABLE TEXTCHANNEL(
TextChannelID VARCHAR(6) NOT NULL,
ServerID VARCHAR(6) NOT NULL,
Name VARCHAR(50),
CreationDate DATE,
CONSTRAINT TEXTCHANNEL_PK PRIMARY KEY(TextChannelID),
CONSTRAINT TEXTCHANNEL_FK FOREIGN KEY(ServerID) REFERENCES SERVER(ServerID));
```

```
CREATE TABLE TEXTCOMMENT(
TextCommentID VARCHAR(6) NOT NULL,
TextChannelID VARCHAR(6) NOT NULL,
EmployeeID VARCHAR(6) NOT NULL,
Comment VARCHAR(255),
CommentTime TIME,
Pinned VARCHAR(1),
CONSTRAINT TEXTCOMMENT_PK PRIMARY KEY(TextCommentID),
CONSTRAINT TEXTCOMMENT_FK1 FOREIGN KEY(TextChannelID) REFERENCES
TEXTCHANNEL(TextChannelID),
CONSTRAINT TEXTCOMMENT_FK2 FOREIGN KEY(EmployeeID) REFERENCES EMPLOYEE(EmployeeID));

CREATE TABLE EMPLOYEETEXTCHANNEL(
EmployeeID VARCHAR(6) NOT NULL,
TextChannelID VARCHAR(6) NOT NULL,
JoinDate DATE,
Muted VARCHAR(1),
CONSTRAINT EMPLOYEETEXTCHANNEL_PK PRIMARY KEY(EmployeeID,TextChannelID),
CONSTRAINT EMPLOYEETEXTCHANNEL_FK1 FOREIGN KEY(EmployeeID) REFERENCES
EMPLOYEE(EmployeeID),
CONSTRAINT EMPLOYEETEXTCHANNEL_FK2 FOREIGN KEY(TextChannelID) REFERENCES
TEXTCHANNEL(TextChannelID));


CREATE TABLE PROJECT(
ProjectID VARCHAR(6) NOT NULL,
ServerID VARCHAR(6) NOT NULL,
Name VARCHAR(50),
Description VARCHAR(255),
CreationDate DATE,
CONSTRAINT PROJECT_PK PRIMARY KEY(ProjectID),
CONSTRAINT PROJECT_FK FOREIGN KEY(ServerID) REFERENCES SERVER(ServerID));

CREATE TABLE TASK(
TaskID VARCHAR(6) NOT NULL,
ProjectID VARCHAR(6) NOT NULL,
Name VARCHAR(50),
Description VARCHAR(255),
CreationDate DATE,
Status VARCHAR(20),
ReminderDate DATE,
CONSTRAINT TASK_PK PRIMARY KEY(TaskID),
CONSTRAINT TASK_FK FOREIGN KEY(ProjectID) REFERENCES PROJECT(ProjectID))

CREATE TABLE TASKCOMMENT(
TaskCommentID VARCHAR(6) NOT NULL,
TaskID VARCHAR(6) NOT NULL,
EmployeeID VARCHAR(6) NOT NULL,
Comment VARCHAR(255),
CommentTime TIME,
Pinned VARCHAR(1),
CONSTRAINT TASKCOMMENT_PK PRIMARY KEY(TaskCommentID),
CONSTRAINT TASKCOMMENT_FK1 FOREIGN KEY(TaskID) REFERENCES TASK(TaskID),
CONSTRAINT TASKCOMMENT_FK2 FOREIGN KEY(EmployeeID) REFERENCES EMPLOYEE(EmployeeID));
```

```
CREATE TABLE EMPLOYEEPROJECT(
EmployeeID VARCHAR(6) NOT NULL,
ProjectID VARCHAR(6) NOT NULL,
JoinDate DATE,
CONSTRAINT EMPLOYEEPROJECT_PK PRIMARY KEY(EmployeeID,ProjectID),
CONSTRAINT EMPLOYEEPROJECT_FK1 FOREIGN KEY(EmployeeID) REFERENCES
EMPLOYEE(EmployeeID),
CONSTRAINT EMPLOYEEPROJECT_FK2 FOREIGN KEY(ProjectID) REFERENCES PROJECT(ProjectID));

CREATE TABLE EMPLOYEETASK(
EmployeeID VARCHAR(6) NOT NULL,
TaskID VARCHAR(6) NOT NULL,
TaskDate DATE,
CONSTRAINT EMPLOYEETASK_PK PRIMARY KEY(EmployeeID,TaskID),
CONSTRAINT EMPLOYEETASK_FK1 FOREIGN KEY(EmployeeID) REFERENCES EMPLOYEE(EmployeeID),
CONSTRAINT EMPLOYEETASK_FK2 FOREIGN KEY(TaskID) REFERENCES TASK(TaskID));


CREATE TABLE SERVERCODES(
ServerCodeID VARCHAR(6) NOT NULL,
ServerID VARCHAR(6) NOT NULL,
CreationTime TIME,
CONSTRAINT SERVERCODES_PK PRIMARY KEY(ServerCodeID),
CONSTRAINT SERVERCODES_FK FOREIGN KEY(ServerID) REFERENCES SERVER(ServerID));

CREATE TABLE EMPLOYEECODES(
EmployeeID VARCHAR(6) NOT NULL,
ServerCodeID VARCHAR(6) NOT NULL,
JoinTime TIME,
CONSTRAINT EMPLOYEECODES_PK PRIMARY KEY(EmployeeID,ServerCodeID),
CONSTRAINT EMPLOYEECODES_FK1 FOREIGN KEY(EmployeeID) REFERENCES EMPLOYEE(EmployeeID),
CONSTRAINT EMPLOYEECODES_FK2 FOREIGN KEY(ServerCodeID) REFERENCES
SERVERCODES(ServerCodeID));
```

Generate a monthly report of all assigned tasks sorted by the project name, reminder date and employee. - Manager

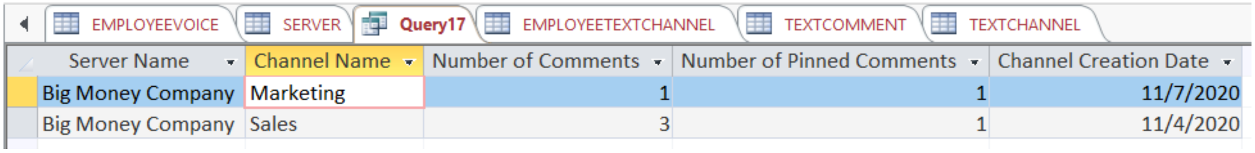
```
SELECT
  SERVER.Name AS `Server Name`, PROJECT.Name AS `Project Name`, PROJECT.Description AS
`Project Description`, TASK.Name AS `Task Name`, TASK.Description AS `Task
Description`, EMPLOYEE.Name AS `Assigned Employee`, TASK.CreationDate AS `Task
Creation Date`, TASK.Status AS `Task Status`, TASK.ReminderDate AS `Task Reminder
Date`
FROM
  (((TASK
INNER JOIN PROJECT
  ON TASK.ProjectID = PROJECT.ProjectID)
INNER JOIN SERVER
  ON PROJECT.ServerID = SERVER.ServerID)
INNER JOIN EMPLOYEETASK
  ON TASK.TaskID = EMPLOYEETASK.TaskID)
INNER JOIN EMPLOYEE
  ON EMPLOYEETASK.EmployeeID = EMPLOYEE.EmployeeID
WHERE
  Month(TASK.CreationDate) = Month(DATE())
GROUP BY
  SERVER.Name, PROJECT.Name, TASK.ReminderDate, PROJECT.Description, TASK.Name,
TASK.Description, EMPLOYEE.Name, TASK.CreationDate, TASK.Status
```



Server Name	Project Name	Project Description	Task Name	Task Description	Assigned Employee	Task Creation	Task Status	Task Reminder Date
Big Money Company	Finance	Finance's tasks and goals.	Mail checks.	sdfasdfffff	Max Price	12/8/2020	Complete	12/11/2020
Big Money Company	Finance	Finance's tasks and goals.	Generate monthly report	dksljbvcksjd	Alice Croft	12/10/2020	Pending	12/27/2020
Big Money Company	Finance	Finance's tasks and goals.	Generate monthly report	dksljbvcksjd	Justine Taft	12/10/2020	Pending	12/27/2020
Big Money Company	Marketing	Marketing's tasks and goals.	Create new slogan.	jhfhdfg	Justine Taft	12/11/2020	Complete	12/21/2020
Big Money Company	Sales	Sales' tasks and goals.	Cold call new customers.	fgvbdf	James Smith	12/13/2020	Pending	12/18/2020
Big Money Company	Sales	Sales' tasks and goals.	Generate sales figures.	fsadsddsfd	Max Price	12/8/2020	Complete	12/24/2020

Generate a monthly report on the activity of text channels. - Manager

```
SELECT
  SERVER.Name AS `Server Name`, TEXTCHANNEL.Name AS `Channel Name`,
COUNT(TEXTCOMMENT.EmployeeID) AS `Number of Comments`, SUM(IIF(TEXTCOMMENT.Pinned =
'T', 1, 0)) AS `Number of Pinned Comments`, TEXTCHANNEL.CreationDate AS `Channel
Creation Date`
FROM
  (TEXTCHANNEL
INNER JOIN TEXTCOMMENT
  ON TEXTCHANNEL.TextChannelID = TEXTCOMMENT.TextChannelID)
INNER JOIN SERVER
  ON TEXTCHANNEL.ServerID = SERVER.ServerID
WHERE
  Month(TEXTCOMMENT.CommentTime) = 11
GROUP BY
  SERVER.Name, TEXTCHANNEL.Name, TEXTCHANNEL.CreationDate
```



Server Name	Channel Name	Number of Comments	Number of Pinned Comments	Channel Creation Date
Big Money Company	Marketing	1	1	11/7/2020
Big Money Company	Sales	3	1	11/4/2020