

# **CSI-3450 Final Project Report**

## **TEC Corporation Database**

### **Grizzlies Team Members:**

Christopher Todd Lair (PHP, HTML, JS Coding, Design Consultant, Webpage Associated Queries, SQL Data page, SQL debugging)

Email: [lair@oakland.edu](mailto:lair@oakland.edu)

Jeffrey Piggott (HTML, Debugging, Tables)

Email: [jpiggott@oakland.edu](mailto:jpiggott@oakland.edu)

Sean Quinn (PHP, HTML, Webpage Queries)

Email: [seanquinn@oakland.edu](mailto:seanquinn@oakland.edu)

Shekia Tillerson (SQL Table Code, SQL Data Code, Data Dictionary)

Email: [shekiatillerson@oakland.edu](mailto:shekiatillerson@oakland.edu)

### **Abstract**

This project realizes the database to be used by the Temporary Employment Corporation (TEC) to post job openings and make informed decisions on the best temporary candidate to hire. The problem this corporation has is the amount of factors involved when searching for eligible candidates. Our database is designed to organize candidates, openings, qualifications, job history, and training sessions currently relevant to that company. Through this database, the company is able to conduct a filtered search into its candidate pool and receive a listed reply of all candidates that could fill that role. Once hired into a position, the database automatically fills the opening and updates that candidate's job history. If the TEC had no database for storing and retrieving this data, the information necessary to make informed decisions would be extremely difficult or impossible. This database is an essential part of the TEC, as it increases their productivity through faster hires and only offering additional training when it is required to fill their opening.

## Entities List:

- Candidates
  - CAN\_NUM(Primary)
  - CAN\_FNAME
  - CAN\_MNAME
  - CAN\_LNAME
  - Q\_CODE
  - JH\_CODE
  - OPEN\_CODE
- Qualifications
  - Q\_CODE(Primary)
  - Q\_DESC
- Job History
  - JH\_CODE(Primary)
  - CAN\_NUM
  - JH\_START\_DATE
  - JH\_END\_DATE
  - JH\_HOURLY\_PAY
  - OPEN\_CODE
- Course
  - CRS\_CODE(Primary)
  - Q\_CODE
  - CAN\_NUM
  - CRS\_FEE
  - CRS\_Q\_REQ
- Training Sessions
  - TS\_CODE(Primary)
  - CRS\_CODE
  - CAN\_NUM
  - TS\_START\_DATE
  - TS\_END\_DATE
- Openings
  - OPEN\_CODE(Primary)
  - Q\_CODE(Primary)
  - CMP\_CODE
  - OPEN\_START\_DATE
  - OPEN\_EST\_END\_DATE
  - OPEN\_HOURLY\_PAY

- Placement
  - OPEN\_CODE
  - CAN\_NUM
  - JH\_CODE
  - PL\_TOTAL\_HOURS
  - CMP\_NUM
- Companies
  - CMP\_CODE(Primary)
  - CMP\_NAME
  - CMP\_NUM
- Qualified Candidates
  - CAN\_NUM(Primary)
  - Q\_CODE(Primary)
- Course Enrollment
  - CAN\_NUM(Primary)
  - CRS\_CODE
- Filled Openings
  - CAN\_NUM(Primary)
  - OPEN\_CODE(Primary)
- Session Enrollment
  - CAN\_NUM(Primary)
  - TS\_CODE(Primary)

## **Business Rules:**

- Any candidate who has worked before has a specific job history. (Naturally, no job history exists if the candidate has never worked.) Each time the candidate works, one additional job history record is created.
- Each candidate has earned several qualifications. Each qualification may be earned by more than one candidate.
- TEC offers courses to help candidates improve their qualifications.
- Some courses cover advanced topics that require specific qualifications as prerequisites. Some courses cover basic topics that do not require any prerequisite qualifications. A course can have several prerequisites. A qualification can be a prerequisite for more than one course.
- Courses are taught during training sessions. A training session is the presentation of a single course. Over time, TEC will offer many training sessions for each course; however, new courses may not have any training sessions scheduled right away.

- Candidates can pay a fee to attend a training session. A training session can accommodate several candidates, although new training sessions will not have any candidates registered at first.
- TEC also has a list of companies that request temporaries.
- Each time a company requests a temporary employee, TEC makes an entry in the Openings folder. That folder contains an opening number, a company name, required qualifications, a starting date, an anticipated ending date, and hourly pay.
- Each opening requires only one specific or main qualification.
- When a candidate matches the qualification, the job is assigned, and an entry is made in the Placement Record folder. The folder contains such information as an opening number, candidate number, and total hours worked. In addition, an entry is made in the job history for the candidate.
- An opening can be filled by many candidates, and a candidate can fill many openings.
- TEC uses special codes to describe a candidate's qualifications for an opening Table P4.13.

# Data Directory

tec\_company

## candidates

Column	Type	Null	Default	Comments
CAN_NUM (Primary)	int(11)	No		
CAN_FNAME	varchar(256)	Yes	NULL	
CAN_MNAME	varchar(256)	Yes	NULL	
CAN_LNAME	varchar(256)	Yes	NULL	
Q_CODE	varchar(30)	Yes	NULL	
JH_CODE	int(11)	Yes	NULL	
OPEN_CODE	int(11)	Yes	NULL	

### Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	CAN_NUM	0	A	No	
JH_CODE	BTREE	No	No	JH_CODE	0	A	Yes	
OPEN_CODE	BTREE	No	No	OPEN_CODE	0	A	Yes	

## companies

Column	Type	Null	Default	Comments
CMP_CODE (Primary)	int(11)	No		
CMP_NAME	varchar(40)	Yes	NULL	
CMP_NUM	int(11)	Yes	NULL	

### Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	CMP_CODE	0	A	No	

## course

Column	Type	Null	Default	Comments
CRS_CODE (Primary)	int(11)	No		
Q_CODE	varchar(30)	Yes	NULL	
CAN_NUM	int(11)	Yes	NULL	
CRS_FEE	int(11)	Yes	NULL	
CRS_Q_REQ	varchar(30)	Yes	NULL	

### Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
---------	------	--------	--------	--------	-------------	-----------	------	---------

8/21/23, 8:39 PM localhost / 127.0.0.1 / tec\_company | phpMyAdmin 5.2.1

## openings

Column	Type	Null	Default	Comments
OPEN_CODE (Primary)	int(11)	No		
Q_CODE (Primary)	varchar(30)	No		
CMP_CODE	int(11)	Yes	NULL	
OPEN_START_DATE	date	Yes	NULL	
OPEN_EST_END_DATE	date	Yes	NULL	
OPEN_HOURLY_PAY	int(11)	Yes	NULL	

### Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	OPEN_CODE	0	A	No	
				Q_CODE	0	A	No	
Q_CODE	BTREE	No	No	Q_CODE	0	A	No	
CMP_CODE	BTREE	No	No	CMP_CODE	0	A	Yes	

## placement

Column	Type	Null	Default	Comments
OPEN_CODE	int(11)	Yes	NULL	
CAN_NUM	int(11)	Yes	NULL	
JH_CODE	int(11)	Yes	NULL	
PL_TOTAL_HOURS	int(11)	Yes	NULL	
CMP_NUM	int(11)	Yes	NULL	

### Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
OPEN_CODE	BTREE	No	No	OPEN_CODE	0	A	Yes	
CAN_NUM	BTREE	No	No	CAN_NUM	0	A	Yes	

## qualifications

Column	Type	Null	Default	Comments
Q_CODE (Primary)	varchar(30)	No		
Q_DESC	varchar(256)	Yes	NULL	

### Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	Q_CODE	12	A	No	

## qualified\_candiates

8/21/23, 8:39 PM

localhost / 127.0.0.1 / tec\_company | phpMyAdmin 5.2.1

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Q_CODE	BTREE	No	No	Q_CODE	0	A	Yes	

## course\_enrolement

Column	Type	Null	Default	Comments
CAN_NUM (Primary)	int(11)	No		
CRS_CODE (Primary)	int(11)	No		

### Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	CAN_NUM	0	A	No	
				CRS_CODE	0	A	No	
CRS_CODE	BTREE	No	No	CRS_CODE	0	A	No	

## filled\_openings

Column	Type	Null	Default	Comments
CAN_NUM (Primary)	int(11)	No		
OPEN_CODE (Primary)	int(11)	No		

### Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	CAN_NUM	0	A	No	
				OPEN_CODE	0	A	No	
OPEN_CODE	BTREE	No	No	OPEN_CODE	0	A	No	

## job\_history

Column	Type	Null	Default	Comments
JH_CODE (Primary)	int(11)	No		
CAN_NUM	int(11)	Yes	NULL	
JH_START_DATE	date	Yes	NULL	
JH_END_DATE	date	Yes	NULL	
JH_HOURLY_PAY	int(11)	Yes	NULL	
OPEN_CODE	int(11)	Yes	NULL	

### Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	JH_CODE	0	A	No	
CAN_NUM	BTREE	No	No	CAN_NUM	0	A	Yes	

8/21/23, 8:39 PM

localhost / 127.0.0.1 / tec\_company | phpMyAdmin 5.2.1

Column	Type	Null	Default	Comments
CAN_NUM (Primary)	int(11)	No		
Q_CODE (Primary)	varchar(30)	No		

### Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	CAN_NUM	0	A	No	
				Q_CODE	0	A	No	
Q_CODE	BTREE	No	No	Q_CODE	0	A	No	

## session\_enrolement

Column	Type	Null	Default	Comments
CAN_NUM (Primary)	int(11)	No		
TS_CODE (Primary)	int(11)	No		

### Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	CAN_NUM	0	A	No	
				TS_CODE	0	A	No	
TS_CODE	BTREE	No	No	TS_CODE	0	A	No	

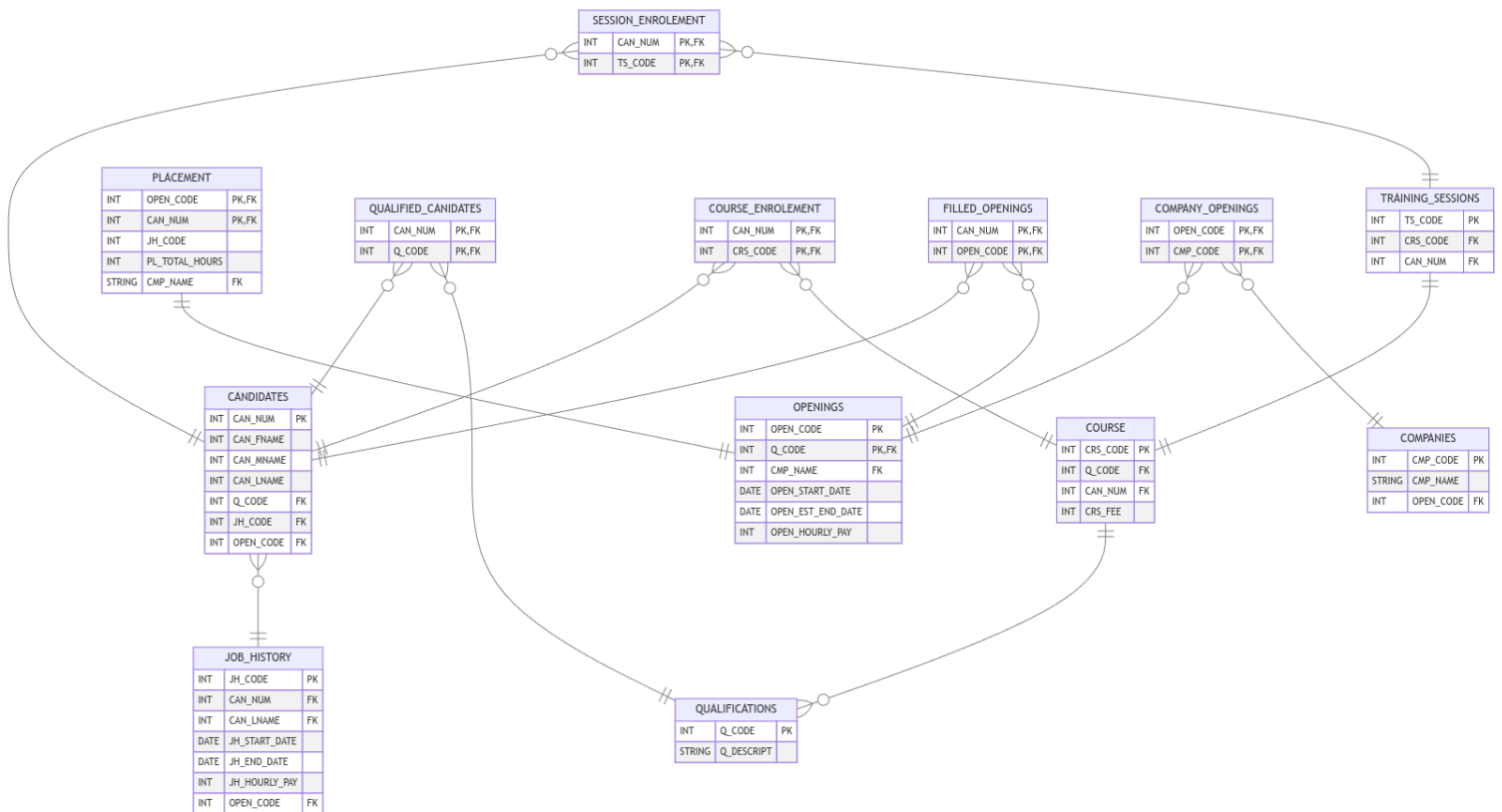
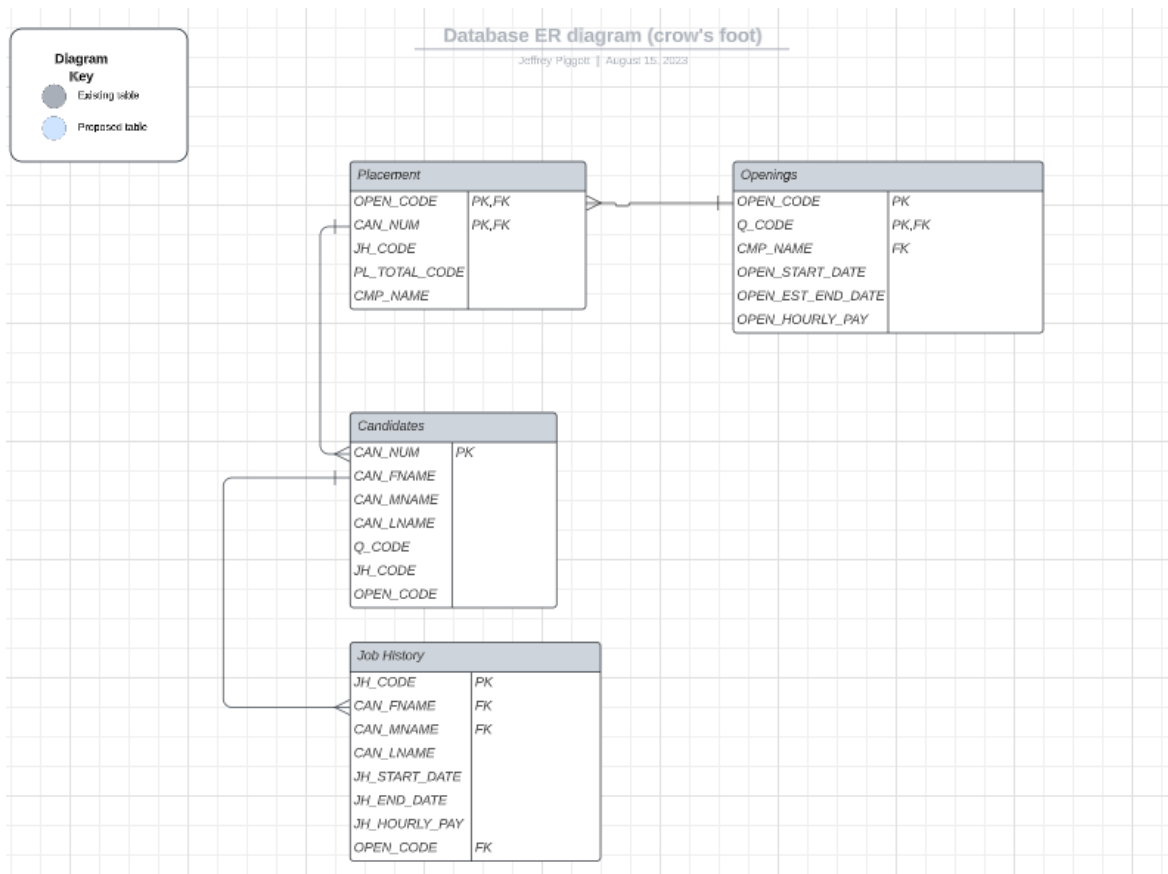
## training\_sessions

Column	Type	Null	Default	Comments
TS_CODE (Primary)	int(11)	No		
CRS_CODE	int(11)	Yes	NULL	
CAN_NUM	int(11)	Yes	NULL	
TS_START_DATE	date	Yes	NULL	
TS_END_DATE	date	Yes	NULL	

### Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	TS_CODE	0	A	No	
CRS_CODE	BTREE	No	No	CRS_CODE	0	A	Yes	
CAN_NUM	BTREE	No	No	CAN_NUM	0	A	Yes	

# Entity Relationship Model



**Table P4.13** Codes for Problem 13

Code	Description
SEC-45	Secretarial work; candidate must type at least 45 words per minute
SEC-60	Secretarial work; candidate must type at least 60 words per minute
CLERK	General clerking work
PRG-PY	Programmer, Python
PRG-C++	Programmer, C++
DBA-ORA	Database Administrator, Oracle
DBA-DB2	Database Administrator, IBM DB2
DBA-SQLSERV	Database Administrator, MS SQL Server
SYS-1	Systems Analyst, level 1
SYS-2	Systems Analyst, level 2
NW-CIS	Network Administrator, Cisco experience
WD-CF	Web Developer, ColdFusion

## Problems Faced and Solutions Found

- Errors with PK's and FK's
  - Went back and fixed data types, referenced entities correctly
- Candidates to Training Sessions was M:M
  - Created bridging entities to make the relationship 1:M

## Forms of Communication

- Github
- Google Suite
- Discord

## Demonstration

The screenshot shows the phpMyAdmin interface for a database named 'TEC\_COMPANY'. The left sidebar displays a tree view of the database structure, including tables like CANDIDATES, COMPANIES, COMPANY\_OPENINGS, COURSE, COURSE\_ENROLEMENT, FILLED\_OPENINGS, JOB\_HISTORY, OPENINGS, PLACEMENT, QUALIFICATIONS, QUALIFIED\_CANDIDATES, SESSION\_ENROLEMENT, and TRAINING\_SESSIONS. The main panel shows the 'Structure' tab for the 'TEC\_COMPANY' database. It lists 13 tables with their respective row counts, types, collations, sizes, and overheads. The 'CANDIDATES' table has 8 rows, 'COMPANIES' has 6, 'COMPANY\_OPENINGS' has 0, 'COURSE' has 11, 'COURSE\_ENROLEMENT' has 0, 'FILLED\_OPENINGS' has 0, 'JOB\_HISTORY' has 4, 'OPENINGS' has 4, 'PLACEMENT' has 0, 'QUALIFICATIONS' has 12, 'QUALIFIED\_CANDIDATES' has 0, 'SESSION\_ENROLEMENT' has 2, and 'TRAINING\_SESSIONS' has 2. The 'Sum' row indicates a total of 49 rows across all tables. The interface also includes a 'Filters' section at the top and a 'Create new table' button at the bottom.

Table	Action	Rows	Type	Collation	Size	Overhead
CANDIDATES	Browse Structure Search Insert Empty Drop	8	InnoDB	utf8mb4_general_ci	48.0 K	18
COMPANIES	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_general_ci	16.0 K	18
COMPANY_OPENINGS	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	32.0 K	18
COURSE	Browse Structure Search Insert Empty Drop	11	InnoDB	utf8mb4_general_ci	64.0 K	18
COURSE_ENROLEMENT	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	32.0 K	18
FILLED_OPENINGS	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	32.0 K	18
JOB_HISTORY	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	48.0 K	18
OPENINGS	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	48.0 K	18
PLACEMENT	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 K	18
QUALIFICATIONS	Browse Structure Search Insert Empty Drop	12	InnoDB	utf8mb4_general_ci	16.0 K	18
QUALIFIED_CANDIDATES	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	32.0 K	18
SESSION_ENROLEMENT	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	32.0 K	18
TRAINING_SESSIONS	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	48.0 K	18
<b>13 tables</b>	<b>Sum</b>	<b>49</b>	<b>InnoDB</b>	<b>utf8mb4_general_ci</b>	<b>496.0 K</b>	<b>0 B</b>

## Demonstration: Candidate Form

### Job History

Show Your Job History

Job History Start Date:

Job History End Date:

Hourly Pay:

Add

### Openings Search

Opening Code:

Opening Code:

Submit

### Course Search

Qualification Code:

Submit

Course Code:

Submit

### Training Session Search

Session Code:

Submit

### Training Session Enroll

Session Code:

Enroll

### Training Session Drop

Session Code:

Drop

## Demonstration: Company Form

### Job History

Job History Start Date:

Job History End Date:

Hourly Pay:

Search

### Openings Search

Opening Code:

Opening Code:

Submit

### Openings Deletion

Opening Code:

Submit



## SQL CODE (Data)

```
-- CANDIDATES
INSERT INTO CANDIDATES (CAN_FNAME, CAN_LNAME) VALUES ('John', 'Romero');
INSERT INTO CANDIDATES (CAN_FNAME, CAN_LNAME) VALUES ('Tom', 'Hall');
INSERT INTO CANDIDATES (CAN_FNAME, CAN_LNAME) VALUES ('Sandy', 'Petersen');
INSERT INTO CANDIDATES (CAN_FNAME, CAN_LNAME) VALUES ('John', 'Carmack');
INSERT INTO CANDIDATES (CAN_FNAME, CAN_LNAME) VALUES ('John', 'Romero');
INSERT INTO CANDIDATES (CAN_FNAME, CAN_LNAME) VALUES ('Dave', 'Taylor');
INSERT INTO CANDIDATES (CAN_FNAME, CAN_LNAME) VALUES ('Adrian', 'Carmack');
INSERT INTO CANDIDATES (CAN_FNAME, CAN_LNAME) VALUES ('Kevin', 'Cloud');
INSERT INTO CANDIDATES (CAN_FNAME, CAN_LNAME) VALUES ('Bobby', 'Prince');

-- COMPANIES
INSERT INTO COMPANIES (CMP_NAME) VALUES ('ID software');
INSERT INTO COMPANIES (CMP_NAME) VALUES ('Raven software');
INSERT INTO COMPANIES (CMP_NAME) VALUES ('Activision');
INSERT INTO COMPANIES (CMP_NAME) VALUES ('GT Software');
INSERT INTO COMPANIES (CMP_NAME) VALUES ('Microsoft');
INSERT INTO COMPANIES (CMP_NAME) VALUES ('ION Storm');

-- OPENINGS
INSERT INTO OPENINGS (Q_CODE, CMP_CODE, OPEN_START_DATE, OPEN_EST_END_DATE,
OPEN_HOURLY_PAY) VALUES ('PRG-C++', '3', '1998-5-12', '1998-9-12', '15');
INSERT INTO OPENINGS (Q_CODE, CMP_CODE, OPEN_START_DATE, OPEN_EST_END_DATE,
OPEN_HOURLY_PAY) VALUES ('PRG-C++', '6', '1998-5-12', '1999-9-12', '17');
INSERT INTO OPENINGS (Q_CODE, CMP_CODE, OPEN_START_DATE, OPEN_EST_END_DATE,
OPEN_HOURLY_PAY) VALUES ('SYS-2', '5', '1999-4-26', '2002-8-7', '40');
INSERT INTO OPENINGS (Q_CODE, CMP_CODE, OPEN_START_DATE, OPEN_EST_END_DATE,
OPEN_HOURLY_PAY) VALUES ('SEC-60', '2', '1995-6-26', '1996-2-16', '9');

-- COURSES
INSERT INTO COURSE (CRS_CODE, Q_CODE, CRS_FEE) VALUES ('2001', 'SEC-45',
'2500');
```

```
INSERT INTO COURSE (Q_CODE, CRS_FEE, CRS_Q_REQ) VALUES ('SEC-60', '3500', 'SEC-45');
```

```
INSERT INTO COURSE (Q_CODE, CRS_FEE) VALUES ('CLERK', '500');
```

```
INSERT INTO COURSE (Q_CODE, CRS_FEE) VALUES ('SYS-1', '1500');
```

```
INSERT INTO COURSE (Q_CODE, CRS_FEE, CRS_Q_REQ) VALUES ('SYS-2', '1500', 'SYS-1');
```

```
INSERT INTO COURSE (Q_CODE, CRS_FEE) VALUES ('PRG-PY', '3000');
```

```
INSERT INTO COURSE (Q_CODE, CRS_FEE) VALUES ('PRG-C++', '3500');
```

```
INSERT INTO COURSE (Q_CODE, CRS_FEE) VALUES ('DBA-ORA', '4500');
```

```
INSERT INTO COURSE (Q_CODE, CRS_FEE) VALUES ('DBA-DB2', '4500');
```

```
INSERT INTO COURSE (Q_CODE, CRS_FEE) VALUES ('DBA-SQLSERV', '4500');
```

```
INSERT INTO COURSE (Q_CODE, CRS_FEE) VALUES ('NW-CIS', '2000');
```

```
INSERT INTO COURSE (Q_CODE, CRS_FEE) VALUES ('WD-CF', '1500');
```

## SQL CODE (Tables)

```
/* Type of SQL : MySQL */  
/* This script file creates the following tables: */  
/* QUALIFICATIONS, COURSE, CANDIDATES, JOB_HISTORY, TRAINING_SESSIONS, */  
/* BRIDGE_CAN_Q, BRIDGE_CAN_CRS, BRIDGE_CAN_TS, BRIDGE_CAN_OPEN, */  
/* BRIDGE_OPEN_CMP, OPENINGS, COMPANIES, SESSION_ENROLLMENT */
```

```
DROP DATABASE IF EXISTS TEC_COMPANY;
```

```
CREATE DATABASE TEC_COMPANY;
```

```
USE TEC_COMPANY;
```

```
/* dropped q_code PK; only attributes w/ int data types can be PK's */
```

```
CREATE TABLE IF NOT EXISTS QUALIFICATIONS(  
Q_CODE VARCHAR(30) PRIMARY KEY,  
Q_DESC VARCHAR(256)
```

```

);

/* data for QUALIFICATIONS table*/
INSERT INTO QUALIFICATIONS VALUES ('SEC-45', 'Secretarial work; candidate must
type at least 45 words per minute');
INSERT INTO QUALIFICATIONS VALUES ('SEC-60', 'Secretarial work; candidate must
type at least 60 words per minute');
INSERT INTO QUALIFICATIONS VALUES ('CLERK', 'General clerkin work');
INSERT INTO QUALIFICATIONS VALUES ('PRG-PY', 'Programmer, Python');
INSERT INTO QUALIFICATIONS VALUES ('PRG-C++', 'Programmer, C++');
INSERT INTO QUALIFICATIONS VALUES ('DBA-ORA', 'Database Administrator,
Oracle');
INSERT INTO QUALIFICATIONS VALUES ('DBA-DB2', 'Database Administrator, IBM
DB2');
INSERT INTO QUALIFICATIONS VALUES ('DBA-SQLSERV', 'Database Administrator, MS
SQL Server');
INSERT INTO QUALIFICATIONS VALUES ('SYS-1', 'Systems Analyst, Level 1');
INSERT INTO QUALIFICATIONS VALUES ('SYS-2', 'Systems Analyst, Level 2');
INSERT INTO QUALIFICATIONS VALUES ('NW-CIS', 'Network Administrator, Cisco
experience');
INSERT INTO QUALIFICATIONS VALUES ('WD-CF', 'Web Developer, ColdFusion');

/* dropped can_lname PK; only attributes w/ int data types can be PK's */
CREATE TABLE IF NOT EXISTS CANDIDATES(
CAN_NUM INT PRIMARY KEY AUTO_INCREMENT,
CAN_FNAME VARCHAR(256),
CAN_MNAME VARCHAR(256),
CAN_LNAME VARCHAR(256),
Q_CODE VARCHAR(30),
JH_CODE INT,
OPEN_CODE INT
);

```

```
CREATE TABLE IF NOT EXISTS COURSE(  
CRS_CODE INT PRIMARY KEY AUTO_INCREMENT,  
Q_CODE VARCHAR(30),  
CAN_NUM INT,  
CRS_FEE INT,  
CRS_Q_REQ VARCHAR(30),  
FOREIGN KEY (CAN_NUM) REFERENCES CANDIDATES (CAN_NUM),  
FOREIGN KEY (Q_CODE) REFERENCES QUALIFICATIONS (Q_CODE),  
FOREIGN KEY (CRS_Q_REQ) REFERENCES QUALIFICATIONS (Q_CODE)  
);
```

```
CREATE TABLE IF NOT EXISTS TRAINING_SESSIONS(  
TS_CODE INT PRIMARY KEY AUTO_INCREMENT,  
CRS_CODE INT,  
CAN_NUM INT,  
TS_START_DATE DATE,  
TS_END_DATE DATE,  
FOREIGN KEY (CRS_CODE) REFERENCES COURSE (CRS_CODE)  
);
```

```
/* dropped ts_code FK; FK'S must refer to a PK in another table. */
```

```
CREATE TABLE IF NOT EXISTS TRAINING_SESSIONS(  
CAN_NUM INT,  
TS_CODE INT PRIMARY KEY,  
FOREIGN KEY (CAN_NUM) REFERENCES CANDIDATES (CAN_NUM),  
FOREIGN KEY (TS_CODE) REFERENCES TRAINING_SESSIONS (TS_CODE)  
);
```

```
CREATE TABLE IF NOT EXISTS QUALIFIED_CANIDATES(  
CAN_NUM INT,  
Q_CODE VARCHAR(30),  
PRIMARY KEY (CAN_NUM, Q_CODE),  
FOREIGN KEY (CAN_NUM) REFERENCES CANDIDATES (CAN_NUM),
```

```

FOREIGN KEY (Q_CODE) REFERENCES QUALIFICATIONS (Q_CODE)
);

CREATE TABLE IF NOT EXISTS COURSE_ENROLEMENT(
CAN_NUM INT,
CRS_CODE INT,
PRIMARY KEY (CAN_NUM, CRS_CODE),
FOREIGN KEY (CAN_NUM) REFERENCES CANDIDATES (CAN_NUM),
FOREIGN KEY (CRS_CODE) REFERENCES COURSE (CRS_CODE)
);

CREATE TABLE IF NOT EXISTS SESSION_ENROLEMENT(
CAN_NUM INT,
TS_CODE INT,
PRIMARY KEY (CAN_NUM, TS_CODE),
FOREIGN KEY (CAN_NUM) REFERENCES CANDIDATES (CAN_NUM),
FOREIGN KEY (TS_CODE) REFERENCES TRAINING_SESSIONS (TS_CODE)
);

CREATE TABLE IF NOT EXISTS OPENINGS(
OPEN_CODE INT AUTO_INCREMENT,
Q_CODE VARCHAR(30) ,
FOREIGN KEY (Q_CODE) REFERENCES QUALIFICATIONS (Q_CODE),
CMP_CODE INT,
-- FOREIGN KEY (CMP_CODE) REFERENCES COMPANIES (CMP_CODE),
OPEN_START_DATE DATE,
OPEN_EST_END_DATE DATE,
OPEN_HOURLY_PAY INT,
PRIMARY KEY (OPEN_CODE, Q_CODE)
);

CREATE TABLE IF NOT EXISTS FILLED_OPENINGS(
CAN_NUM INT,

```

```

OPEN_CODE INT,
PRIMARY KEY (CAN_NUM, OPEN_CODE),
FOREIGN KEY (CAN_NUM) REFERENCES CANDIDATES (CAN_NUM),
FOREIGN KEY (OPEN_CODE) REFERENCES OPENINGS (OPEN_CODE)
);

/* dropped can_lname FK; FK'S must refer to a PK in another table. */
CREATE TABLE IF NOT EXISTS JOB_HISTORY(
JH_CODE INT PRIMARY KEY AUTO_INCREMENT,
CAN_NUM INT,
JH_START_DATE DATE,
JH_END_DATE DATE,
JH_HOURLY_PAY INT,
OPEN_CODE INT,
FOREIGN KEY (CAN_NUM) REFERENCES CANDIDATES (CAN_NUM),
FOREIGN KEY (OPEN_CODE) REFERENCES OPENINGS (OPEN_CODE)
);

/* added cmp_name attribute */
/* dropped cmp_name PK; only attributes w/ int data types can be PK's */
CREATE TABLE IF NOT EXISTS COMPANIES(
CMP_CODE INT PRIMARY KEY AUTO_INCREMENT,
CMP_NAME VARCHAR(40),
CMP_NUM INT
);

/* dropped cmp_name FK; FK'S must refer to a PK in another table. */
CREATE TABLE IF NOT EXISTS PLACEMENT(
OPEN_CODE INT,
CAN_NUM INT,
JH_CODE INT,
PL_TOTAL_HOURS INT,
CMP_NUM INT,

```

```
FOREIGN KEY (OPEN_CODE) REFERENCES OPENINGS (OPEN_CODE),
FOREIGN KEY (CAN_NUM) REFERENCES CANDIDATES (CAN_NUM)
);

ALTER TABLE CANDIDATES
ADD FOREIGN KEY (JH_CODE) REFERENCES JOB_HISTORY (JH_CODE),
ADD FOREIGN KEY (OPEN_CODE) REFERENCES OPENINGS (OPEN_CODE),
AUTO_INCREMENT = 1001;

ALTER TABLE JOB_HISTORY
ADD FOREIGN KEY (OPEN_CODE) REFERENCES OPENINGS (OPEN_CODE),
AUTO_INCREMENT = 4001;

ALTER TABLE OPENINGS
ADD FOREIGN KEY (CMP_CODE) REFERENCES COMPANIES (CMP_CODE);

ALTER TABLE TRAINING_SESSIONS
ADD FOREIGN KEY (CAN_NUM) REFERENCES CANDIDATES (CAN_NUM);
AUTO_INCREMENT = 3001;

ALTER TABLE QUALIFIED_CANIDATES
ADD FOREIGN KEY (Q_CODE) REFERENCES QUALIFICATIONS (Q_CODE);

ALTER TABLE COURSE
ADD FOREIGN KEY (Q_CODE) REFERENCES QUALIFICATIONS (Q_CODE),
AUTO_INCREMENT = 2001;
```