**Final Project Report**

**Woodbridge Church Members Database (WCMD)**

**Kennedy Donkor**

**DBST 651, Fall 2020**

**11/03/2020**

**Statement of Work**

**Overview:**

The Woodbridge Ghanaian Seventh-Day Adventist Church is a local Seventh-Day Adventist church that serves members who are mainly from Ghana and of Ghanaian descent in the Northern Virginia area. They are a community of believers that are seeking to help each other mentally, physically, and spiritually while also serving the community at large. The church is organized into different departments with each department serving a specific role in the church. Currently the departments with its leadership are all recorded manually and distributed to the various departmental leaders. Sometimes it is very difficult to find the members of a department including its leaders especially when one loses the paper copy. This Statement of Work (SOW) will expand on how a database can be created to help solve this problem.

**Executive Summary:**

A database created for Woodbridge Ghanaian Seventh-Day Adventist Church will increase efficiency, productivity, and impact of the church on its members and the community at large.

Creating an electronic database will serve this congregation by helping with the streamline and distribution of information. Projects are delayed or sometimes not implemented at all due to a lack of communication. Having the right information delivered to the concerned party becomes difficult as the members are not sure who is in charge of what. Once this database is setup, it can increase our communication with the respective conference officials in order to take advantage of all the programs available to the church and its community. Not a lot of storage space will be required for the project and the manpower needed to complete this project will be minimal which will bring the cost of the overall project down.

**Objectives:**

This database will help in the construction and maintenance of an online source of information for the immediate retrieval for the members of the church and the community as a whole. Church members will be able to search for different departments using a variety of measures. Using information like the assigned identification for the department, the department name, or one of the members of the department, they should be able to pull up information needed for the department. Once the user pulls up the department of choice, they should be able to access every information about the department including the department leader and the members of the department. The database will be simple enough for even the non-technology savvy individuals to use.

**Project Scope:**

**Include in Project Scope:**

Collecting the data requirements from the church

Gain access to conference information

Installing required software

Creating Business rules

Generating the Entity-Relationship model

Creating the Data Definition Languages (DDL)

Creating the Data Manipulation Language (DML)

Performing data validation checks and the addition of constraints

**Excluded from Project Scope:**

Maintenance of the database

Using systems that are compatible with the database

Database system that is compatible with the conference database

**Database Goals, Expectations, and Deliverables:**

The goal of the database is to organize the departments of the church along with the leaders and the members of the department for easy access. The database is expected to have all the details of each department and its leadership all the way up to the conference level. Having the hierarchy of the departments listed along with their contact information. The deliverables will include the database, the entity-attribute list, the data definition list, the data model, the DDL file.

**Database Benefits:**

Currently there exist no church database. When one wants to find an information for a department, they either have to consult the paper copy which is only made available to the departmental leaders or call someone. This is not very efficient and hinders the progress of the church in terms of decimating information through the proper channels and seeing results in a timely manner or sometimes at all. This will make the information readily available to the concerned department. It will decrease response and implementation time making the church more effective in terms of reaching its members and community. It will decrease misinformation and information getting into the wrong hands by having information delivered to the concerned department directly. It will also increase the communication between the local church departmental leaders and the conference departmental leaders.

**Project Hardware and Software Tools:**

This project will be completed within the Virtual Machine environment provided by University of Maryland Global Campus. The virtual environment will be accessed from a MacBook Air with macOS Catalina version 10.15.6 processor. The virtual environment PC is AUV-UMGCVDA-33. The project will also be using E-R Assistant and Oracle SQL Developer version 19.1.0.094.

**SQL Usage and Style Guide:**

Consistent and descriptive identifiers and names will be used. Example, all names will be descriptive and begin with a letter and if a name is more than one word an underscore will be used. For example, “First Name” will be written as “First\_Name”. The ISO 8601 compliant date formation will be used and inputted as YYYY-MM-DD. When comments are use, they will open with /\* and close with \*/. Adequate white space will be used and statements broken up over multiple lines to make it more legible. If a comment is required after a SQL statement then “--“ will be used as the beginning of the comment. Hungarian notations will be avoided along with plurals as much as possible.

**Requirements Definition Document**

**Entity Relationship/Business Rules**

Entity: Position – this refers to the positions that are to be filled in each department by the church members. For each department, there is a director, an assistant director, a secretary, and a treasurer. The PositionID will be created generically to identify the position.

Primary Key: PositionID

Foreign Key: N/A

Relationship: “Holds” between Church Member and Position

Cardinality: 1:M between Position and Church Member

Business Rule: Positions are held by many church members but each position can only be held by one church member. Each position must be held by a church member but not all church members hold positions.

Entity: Church member – this refers to all the members of the church. The MemberID will also be created generically to identify each member. The members name, email address, and phone numbers will also be collected as part of their records and can be used to search for them besides their MemberID. The gender of the church member is also identified for census purposes and what position they hold in the church if any and how long they will be serving in that position.

Primary Key: MemberID

Foreign Key: PositionID and DepartmentID

Relationship: “Belongs” between Church Member and Department

Cardinality: M:N between Church Member and Department

Business Rule: Each church member can belong to many departments and each department can have many church members in them. Each department must have some church members but not all church members belong to a department.

Entity: Department – this refers to all the departments that function in the church. There is a department ID that identifies the department that will also be generically created. Each department has a name and in what capacity they function. The department’s contact information is also made available so everyone knows where they can be reached. In the department’s information is the elder in charge, the conference leader of that department and the members serving in that department.

Primary Key: DepartmentID

Foreign Key: ElderID, ConferenceLeaderID, and MemberID

Relationship: “Oversees” between Department and Elder

Cardinality: 1:M between Elder and Department

Business Rule: One elder can oversee many departments but each department can only be overseen by one elder. A department has to have an overseer and an elder has to oversee a department.

Entity: Elder – these are the selected elders of the church who are to work closely with the church pastor in the running of the church. Their office information and the department or departments that are in charge of are also listed.

Primary Key: ElderID

Foreign Key: DepartmentID

Relationship: “Reports” between Elder and Pastor

Cardinality: 1:M between Pastor and Elder

Business Rule: Many elders can report to the pastor but only one pastor can be reported to. The elders must report to the pastor and the pastor must be reported to.

Entity: Pastor – the pastor serves as the general overseer of the whole church. The pastor’s office information is also collected. The pastor does not have a term limit and therefore that was not included

Primary Key: PastorID

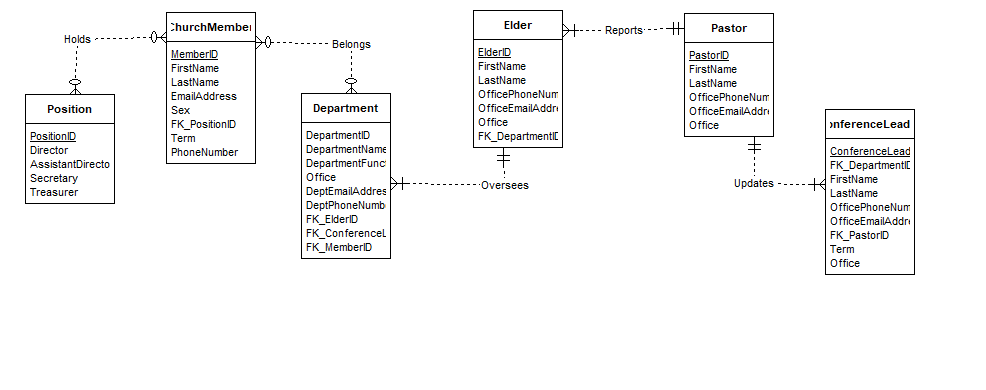
Foreign Key: ConferenceLeaderID

Relationship: “Updates” between Pastor and Conference Leader

Cardinality: 1:M between Pastor and Conference Leader

Business Rule: the pastor updates the many conference leaders of programs being implemented by the departments and the conference leader updates the church pastor of any new programs the conference wants to be implemented. The pastor must update the conference leaders and the conference leaders must update the pastor.

**Entity Relationship Diagram**



**DDL Source Code**

/\*DROP TABLE POSITION CASCADE CONSTRAINTS;

DROP TABLE CHURCH\_MEMBER CASCADE CONSTRAINTS;

DROP TABLE DEPARTMENT CASCADE CONSTRAINTS;

DROP TABLE ELDER CASCADE CONSTRAINTS;

DROP TABLE PASTOR CASCADE CONSTRAINTS;

DROP TABLE CONFERENCE\_LEADER CASCADE CONSTRAINTS;

DROP SEQUENCE seq\_position\_id;

DROP SEQUENCE seq\_church\_member\_id;

DROP SEQUENCE seq\_department\_id;

DROP SEQUENCE seq\_elder\_id;

DROP SEQUENCE seq\_pastor\_id;

DROP SEQUENCE seq\_conference\_leader\_id;

\*/

--THE TYPE OF POSITIONS IN THE CHURCH THAT CHURCH MEMBERS HOLD--

CREATE TABLE POSITION(POSITION\_ID VARCHAR(10) PRIMARY KEY,

DIRECTOR VARCHAR(20) NOT NULL,

ASSISTANT\_DIRECTOR VARCHAR(20) NOT NULL,

SECRETARY VARCHAR(20) NOT NULL,

TREASURER VARCHAR(20)

);

--THESE ARE THE BAPTIZED MEMBERS OF THE CHURCH AND SOME HOLD POSITIONS IN DIFFERENT DEPARTMENTS--

CREATE TABLE CHURCH\_MEMBER(MEMBER\_ID INT PRIMARY KEY,

FIRST\_NAME CHAR(20) NOT NULL,

LAST\_NAME CHAR(20) NOT NULL,

EMAIL\_ADDRESS VARCHAR(20) NOT NULL,

SEX CHAR(1) NOT NULL, -- M or F for census purposes

TERM DATE, --The length of service before the next election cycle

PHONE\_NUMBER INT,

POSITION\_ID VARCHAR(10), --The position held by the church member

CONSTRAINT FK\_MEMBER\_POSITION\_ID FOREIGN KEY (POSITION\_ID) REFERENCES POSITION(POSITION\_ID)

);

CREATE INDEX IX\_MEMBER\_POSITION ON CHURCH\_MEMBER(POSITION\_ID);

--THE DEPARTMENTS THAT OPERATE IN THE CHURCH--

CREATE TABLE DEPARTMENT(DEPARTMENT\_ID INT PRIMARY KEY,

DEPARTMENT\_NAME CHAR(20) NOT NULL,

DEPARTMENT\_FUNCTION VARCHAR(100) NOT NULL, --The function of the department

OFFICE VARCHAR(5) NOT NULL, --The department's office room number in the church

DEPT\_EMAIL\_ADDRESS VARCHAR(20) NOT NULL,

DEPT\_PHONE\_NUMBER INT NOT NULL, --The phone number specifically for the department

ELDER\_ID INT, --The elder in charge of the department

CONSTRAINT FK\_DEPT\_ELDER\_ID FOREIGN KEY (ELDER\_ID) REFERENCES ELDER(ELDER\_ID),

CONFERENCE\_LEADER\_ID VARCHAR(20),

CONSTRAINT FK\_DEPT\_CONFERENCE\_LEADER\_ID FOREIGN KEY (CONFERENCE\_LEADER\_ID) REFERENCES CONFERENCE\_LEADER(CONFERENCE\_LEADER\_ID),

MEMBER\_ID INT, --The members of the department

CONSTRAINT FK\_DEPT\_MEMBER\_ID FOREIGN KEY (MEMBER\_ID) REFERENCES CHURCH\_MEMBER(MEMBER\_ID)

);

CREATE INDEX IX\_DEPT\_ELDER ON DEPARTMENT(ELDER\_ID);

CREATE INDEX IX\_DEPT\_CONFERENCE\_LEADER ON DEPARTMENT(CONFERENCE\_LEADER\_ID);

CREATE INDEX IX\_DEPT\_MEMBERS ON DEPARTMENT(MEMBER\_ID);

--THE ELECTED ELDERS OF THE CHURCH OVERSEEING DIFFERENT DEPARTMENTS--

CREATE TABLE ELDER(ELDER\_ID INT PRIMARY KEY,

FIRST\_NAME CHAR(20) NOT NULL,

LAST\_NAME CHAR(20) NOT NULL,

OFFICE\_PHONE\_NUMBER NUMERIC(10) NOT NULL, --the elder's office phone number

OFFICE\_EMAIL\_ADDRESS VARCHAR(20) NOT NULL, --email address for the office of the elder

OFFICE VARCHAR(5) NOT NULL, --the elders office room number in the church

DEPARTMENT\_ID INT); --the department the elder oversees

CREATE INDEX IX\_ELDER\_DEPT ON ELDER(DEPARTMENT\_ID);

--ADDING FOREIGN KEY CONSTRAINT ON AN EXISTING TABLE--

ALTER TABLE ELDER

ADD CONSTRAINT FK\_ELDER\_DEPARTMENT\_ID FOREIGN KEY (DEPARTMENT\_ID) REFERENCES DEPARTMENT(DEPARTMENT\_ID)

;

--THE CHURCH PASTOR WHO RECEIVES REPORTS FROM THE ELDERS AND UPDATE THE CONFERENCE LEADERS--

CREATE TABLE PASTOR(PASTOR\_ID VARCHAR(20) PRIMARY KEY,

FIRST\_NAME VARCHAR(20) NOT NULL,

LAST\_NAME VARCHAR(20) NOT NULL,

OFFICE\_PHONE\_NUMBER NUMERIC(10) NOT NULL,

OFFICE\_EMAIL\_ADDRESS VARCHAR(20) NOT NULL,

OFFICE VARCHAR(5) NOT NULL --the pastor's office room number

);

--THE CONFERENCE LEADERS OF THE DIFFERENT DEPARTMENTS UPDATES THE LOCAL CHURCH PASTORS OF UPCOMING EVENTS--

CREATE TABLE CONFERENCE\_LEADER(CONFERENCE\_LEADER\_ID VARCHAR(20) PRIMARY KEY,

FIRST\_NAME VARCHAR(20) NOT NULL,

LAST\_NAME VARCHAR(20) NOT NULL,

OFFICE\_PHONE\_NUMBER NUMERIC(10) NOT NULL,

OFFICE\_EMAIL\_ADDRESS VARCHAR(20) NOT NULL,

PASTOR\_ID VARCHAR(20), --the pastor the conference leader reports to

CONSTRAINT FK\_CONFERENCE\_LEADER\_PASTOR\_ID FOREIGN KEY (PASTOR\_ID) REFERENCES PASTOR(PASTOR\_ID),

OFFICE VARCHAR(6) NOT NULL, --their office in the conference building

DEPARTMENT\_ID INT, --the department they are in charge of

TERM DATE --how long they serving for

);

--ADDING FOREIGN KEY CONSTRAINT TO AN EXISTING TABLE--

ALTER TABLE CONFERENCE\_LEADER

ADD CONSTRAINT FK\_CONFERENCE\_LEADER\_DEPARTMENT\_ID FOREIGN KEY (DEPARTMENT\_ID) REFERENCES DEPARTMENT(DEPARTMENT\_ID);

CREATE INDEX IX\_CONFERENCE\_LEADER\_PASTOR ON CONFERENCE\_LEADER(PASTOR\_ID);

CREATE INDEX IX\_CONFERENCE\_LEADER\_DEPT ON CONFERENCE\_LEADER(DEPARTMENT\_ID);

--THIS WILL SERVE AS A SNAPSHOT OF THE DEPT WITH THE DEPTS BASIC INFORMATION INCLUDING THE ELDER OVERSEEING THAT DEPT--

CREATE VIEW DEPARTMENT\_INFO AS

SELECT DEPARTMENT\_NAME, OFFICE, DEPT\_PHONE\_NUMBER, ELDER\_ID FROM DEPARTMENT;

--THIS VIEW GIVES THE CHURCH MEMBERS THE OPPORTUNITY TO STAY IN CONTACT WITH ONE ANOTHER--

CREATE VIEW MEMBERS\_OF\_THE\_CHURCH AS

SELECT FIRST\_NAME, LAST\_NAME, PHONE\_NUMBER FROM CHURCH\_MEMBER;

--IN CASE A MEMBER NEEDS TO REACH OUT TO CONFERENCE LEADERSHIP, THIS VIEW WILL PROVIDE THAT THEIR INFO--

CREATE VIEW CONFERENCE\_INFO AS

SELECT DEPARTMENT\_ID, FIRST\_NAME, LAST\_NAME, OFFICE\_EMAIL\_ADDRESS FROM CONFERENCE\_LEADER;

CREATE SEQUENCE SEQ\_POSITION\_ID;

CREATE SEQUENCE SEQ\_MEMBER\_ID;

CREATE SEQUENCE SEQ\_DEPARTMENT\_ID;

CREATE SEQUENCE SEQ\_ELDER\_ID;

CREATE SEQUENCE SEQ\_PASTOR\_ID;

CREATE SEQUENCE SEQ\_CONFERENCE\_LEADER\_ID;

--THE PURPOSE OF THIS TRIGGER IS TO MAKE SURE WHEN A NEW MEMBER IS ADDED TO THE CHURCH, THEY RECEIVE A MEMBER ID--

CREATE OR REPLACE TRIGGER MEMBERID

BEFORE INSERT ON CHURCH\_MEMBER

FOR EACH ROW

BEGIN

IF :NEW.MEMBER\_ID IS NULL THEN

:NEW.MEMBER\_ID := SEQ\_MEMBER\_ID.NEXTVAL;

END IF;

END;

/

/\*ONCE A CHURCH MEMBER IS NO LONGER SERVING IN A POSITION THIS TRIGGER WILL INFORM THE INQUIRER THE MEMBER IS

NO LONGER SERVING IN AN OFFICIAL CAPACITY\*/

CREATE OR REPLACE TRIGGER POSITIONID

BEFORE UPDATE ON CHURCH\_MEMBER

FOR EACH ROW

BEGIN

IF :NEW.POSITION\_ID IS NULL THEN

:NEW.POSITION\_ID := 'NO OFFICIAL POSITION';

END IF;

END;

/

/\*CHURCH ELDERS ARE ROTATED AFTER A FEW YEARS OF SERVING AND THIS TRIGGER WILL INFORM THE INQUIRER IF

THE ELDER IN QUESTION IS CURRENTLY SERVING\*/

CREATE OR REPLACE TRIGGER CHURCH\_ELDERS

BEFORE UPDATE ON ELDER

FOR EACH ROW

BEGIN

IF :NEW.ELDER\_ID IS NULL THEN

:NEW.ELDER\_ID := 'NOT CURRENTLY A SERVIING ELDER';

END IF;

END;

/

SELECT \* FROM POSITION;

SELECT \* FROM CHURCH\_MEMBER;

SELECT \* FROM DEPARTMENT;

SELECT \* FROM ELDER;

SELECT \* FROM PASTOR;

SELECT \* FROM CONFERENCE\_LEADER;

**DML and Query Source Code**

--Inserting church members information into the database--

INSERT INTO CHURCH\_MEMBER(MEMBER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL\_ADDRESS, SEX, TERM, PHONE\_NUMBER)

VALUES(0001, 'WILL', 'SMITH', 'WILLSMITH@GMAIL.COM', 'M', TO\_DATE('01-JAN-21', 'DD-MON-YY'), '5715721111');

INSERT INTO CHURCH\_MEMBER(MEMBER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL\_ADDRESS, SEX, TERM, PHONE\_NUMBER)

VALUES(0002, 'PHILIP', 'BANKS', 'PBANKS@GMAIL.COM', 'M', TO\_DATE('01-JAN-21', 'DD-MON-YY'), '5715721122');

INSERT INTO CHURCH\_MEMBER(MEMBER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL\_ADDRESS, SEX, TERM, PHONE\_NUMBER)

VALUES(0003, 'CARLTON', 'BANKS', 'CBANKS@GMAIL.COM', 'M', TO\_DATE('01-JAN-21', 'DD-MON-YY'), '5715721123');

INSERT INTO CHURCH\_MEMBER(MEMBER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL\_ADDRESS, SEX, TERM, PHONE\_NUMBER)

VALUES(0004, 'VIVIAN', 'BANKS', 'VBANKS@GMAIL.COM', 'F', TO\_DATE('01-JAN-21', 'DD-MON-YY'), '5715721124');

INSERT INTO CHURCH\_MEMBER(MEMBER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL\_ADDRESS, SEX, TERM, PHONE\_NUMBER)

VALUES(0005, 'HILLARY', 'BANKS', 'HBANKS@GMAIL.COM', 'F', TO\_DATE('01-JAN-21', 'DD-MON-YY'), '5715721125');

INSERT INTO CHURCH\_MEMBER(MEMBER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL\_ADDRESS, SEX, TERM, PHONE\_NUMBER)

VALUES(0006, 'ASHLEY', 'BANKS', 'ABANKS@GMAIL.COM', 'F', TO\_DATE('01-JAN-21', 'DD-MON-YY'), '5715721126');

INSERT INTO CHURCH\_MEMBER(MEMBER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL\_ADDRESS, SEX, TERM, PHONE\_NUMBER)

VALUES(0007, 'NICHOLAS', 'BANKS', 'NBANKS@GMAIL.COM', 'M', TO\_DATE('01-JAN-21', 'DD-MON-YY'), '5715721127');

INSERT INTO CHURCH\_MEMBER(MEMBER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL\_ADDRESS, SEX, TERM, PHONE\_NUMBER)

VALUES(0008, 'JEFFREY', 'BUTLER', 'JBUTLER@GMAIL.COM', 'M', TO\_DATE('01-JAN-21', 'DD-MON-YY'), '5715721128');

INSERT INTO CHURCH\_MEMBER(MEMBER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL\_ADDRESS, SEX, TERM, PHONE\_NUMBER)

VALUES(0009, 'VY', 'SMITH', 'VYSMITH@GMAIL.COM', 'F', TO\_DATE('01-JAN-21', 'DD-MON-YY'), '5715721112');

INSERT INTO CHURCH\_MEMBER(MEMBER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL\_ADDRESS, SEX, TERM, PHONE\_NUMBER)

VALUES(0010, 'HELEN', 'SMITH', 'HSMITH@GMAIL.COM', 'F', TO\_DATE('01-JAN-21', 'DD-MON-YY'), '5715721113');

COMMIT;

ALTER TABLE CHURCH\_MEMBER ADD POSITION CHAR(30);

DELETE FROM CHURCH\_MEMBER WHERE POSITION='DIRECTOR';

UPDATE CHURCH\_MEMBER SET POSITION\_ID = 'M01' WHERE CHURCH\_MEMBER.FIRST\_NAME='WILL';

UPDATE CHURCH\_MEMBER SET POSITION\_ID='T01' WHERE CHURCH\_MEMBER.FIRST\_NAME='CARLTON';

UPDATE CHURCH\_MEMBER SET POSITION\_ID='S01' WHERE CHURCH\_MEMBER.FIRST\_NAME='ASHLEY';

UPDATE CHURCH\_MEMBER SET POSITION\_ID='Y01' WHERE CHURCH\_MEMBER.FIRST\_NAME='JEFFREY';

UPDATE CHURCH\_MEMBER SET POSITION\_ID='CPMO1' WHERE CHURCH\_MEMBER.FIRST\_NAME='PHILIP';

UPDATE CHURCH\_MEMBER SET POSITION\_ID='Y01' WHERE CHURCH\_MEMBER.FIRST\_NAME='NICHOLAS';

UPDATE CHURCH\_MEMBER SET POSITION\_ID='CPMO1' WHERE CHURCH\_MEMBER.FIRST\_NAME='VY';

UPDATE CHURCH\_MEMBER SET POSITION\_ID='CPMO1' WHERE CHURCH\_MEMBER.FIRST\_NAME='HELEN';

UPDATE CHURCH\_MEMBER SET POSITION\_ID='T01' WHERE CHURCH\_MEMBER.FIRST\_NAME='HILLARY';

UPDATE CHURCH\_MEMBER SET POSITION\_ID='T01' WHERE CHURCH\_MEMBER.FIRST\_NAME='VIVIAN';

--Inserting conference leaders information into the database--

INSERT INTO CONFERENCE\_LEADER(CONFERENCE\_LEADER\_ID, FIRST\_NAME, LAST\_NAME, OFFICE\_PHONE\_NUMBER, OFFICE\_EMAIL\_ADDRESS, OFFICE, TERM)

VALUES('CT01', 'JAZZY', 'JEFF', '7037301111', 'JJEFF@GMAIL.COM', 'C200', TO\_DATE('01-JAN-21', 'DD-MON-YY'));

INSERT INTO CONFERENCE\_LEADER(CONFERENCE\_LEADER\_ID, FIRST\_NAME, LAST\_NAME, OFFICE\_PHONE\_NUMBER, OFFICE\_EMAIL\_ADDRESS, OFFICE, TERM)

VALUES('CS01', 'JACKIE', 'AMES', '7037301112', 'JAMES@GMAIL.COM', 'C201', TO\_DATE('01-JAN-21', 'DD-MON-YY'));

INSERT INTO CONFERENCE\_LEADER(CONFERENCE\_LEADER\_ID, FIRST\_NAME, LAST\_NAME, OFFICE\_PHONE\_NUMBER, OFFICE\_EMAIL\_ADDRESS, OFFICE, TERM)

VALUES('CPM01', 'TREVOR', 'COLLINS', '7037301113', 'TCOLLINS@GMAIL.COM', 'C202', TO\_DATE('01-JAN-21', 'DD-MON-YY'));

INSERT INTO CONFERENCE\_LEADER(CONFERENCE\_LEADER\_ID, FIRST\_NAME, LAST\_NAME, OFFICE\_PHONE\_NUMBER, OFFICE\_EMAIL\_ADDRESS, OFFICE, TERM)

VALUES('CM01', 'LISA', 'WILKES', '7037301114', 'LWILKES@GMAIL.COM', 'C203', TO\_DATE('01-JAN-21', 'DD-MON-YY'));

INSERT INTO CONFERENCE\_LEADER(CONFERENCE\_LEADER\_ID, FIRST\_NAME, LAST\_NAME, OFFICE\_PHONE\_NUMBER, OFFICE\_EMAIL\_ADDRESS, OFFICE, TERM)

VALUES('CY01', 'JANICE', 'SMITH', '7037301115', 'JSMITH@GMAIL.COM', 'C204', TO\_DATE('01-JAN-21', 'DD-MON-YY'));

COMMIT;

UPDATE CONFERENCE\_LEADER SET DEPARTMENT\_ID='100', PASTOR\_ID='P100' WHERE CONFERENCE\_LEADER.FIRST\_NAME='JAZZY';

UPDATE CONFERENCE\_LEADER SET DEPARTMENT\_ID='200', PASTOR\_ID='P100' WHERE CONFERENCE\_LEADER.FIRST\_NAME='JACKIE';

UPDATE CONFERENCE\_LEADER SET DEPARTMENT\_ID='300', PASTOR\_ID='P100' WHERE CONFERENCE\_LEADER.FIRST\_NAME='TREVOR';

UPDATE CONFERENCE\_LEADER SET DEPARTMENT\_ID='400', PASTOR\_ID='P100' WHERE CONFERENCE\_LEADER.FIRST\_NAME='LISA';

UPDATE CONFERENCE\_LEADER SET DEPARTMENT\_ID='500', PASTOR\_ID='P100' WHERE CONFERENCE\_LEADER.FIRST\_NAME='JANICE';

--Inserting the different departments information into the database--

INSERT INTO DEPARTMENT(DEPARTMENT\_ID, DEPARTMENT\_NAME, DEPARTMENT\_FUNCTION, OFFICE, DEPT\_EMAIL\_ADDRESS, DEPT\_PHONE\_NUMBER)

VALUES('100', 'TREASURY DEPT', 'TAKE CARE OF THE CHURCH FINANCES', 'A100', 'TDEPT@GMAIL.COM', '7037302222');

INSERT INTO DEPARTMENT(DEPARTMENT\_ID, DEPARTMENT\_NAME, DEPARTMENT\_FUNCTION, OFFICE, DEPT\_EMAIL\_ADDRESS, DEPT\_PHONE\_NUMBER)

VALUES('200', 'SECRETARY DEPT', 'TAKE CARE OF CHURCH RECORDS', 'A200', 'SDEPT@GMAIL.COM', '7037302223');

INSERT INTO DEPARTMENT(DEPARTMENT\_ID, DEPARTMENT\_NAME, DEPARTMENT\_FUNCTION, OFFICE, DEPT\_EMAIL\_ADDRESS, DEPT\_PHONE\_NUMBER)

VALUES('300', 'PERSONAL MIN DEPT', 'TAKE CARE OF CHURCH EVANGELISM', 'A300', 'PMINDEPT@GMAIL.COM', '7037302224');

INSERT INTO DEPARTMENT(DEPARTMENT\_ID, DEPARTMENT\_NAME, DEPARTMENT\_FUNCTION, OFFICE, DEPT\_EMAIL\_ADDRESS, DEPT\_PHONE\_NUMBER)

VALUES('400', 'MUSIC DEPT', 'TAKE CARE OF EVERYTHING MUSIC RELATED', 'A400', 'MDEPT@GMAIL.COM', '7037302225');

INSERT INTO DEPARTMENT(DEPARTMENT\_ID, DEPARTMENT\_NAME, DEPARTMENT\_FUNCTION, OFFICE, DEPT\_EMAIL\_ADDRESS, DEPT\_PHONE\_NUMBER)

VALUES('500', 'YOUTH DEPT', 'TAKE CARE OF THE CHURCH YOUTH', 'A500', 'YDEPT@GMAIL.COM', '7037302226');

COMMIT;

UPDATE DEPARTMENT SET ELDER\_ID='1', MEMBER\_ID='3' WHERE DEPARTMENT\_ID='100';

UPDATE DEPARTMENT SET ELDER\_ID='1', MEMBER\_ID='6' WHERE DEPARTMENT\_ID='200';

UPDATE DEPARTMENT SET ELDER\_ID='2', MEMBER\_ID='2' WHERE DEPARTMENT\_ID='300';

UPDATE DEPARTMENT SET ELDER\_ID='2', MEMBER\_ID='1' WHERE DEPARTMENT\_ID='400';

UPDATE DEPARTMENT SET ELDER\_ID='3', MEMBER\_ID='8' WHERE DEPARTMENT\_ID='500';

--Inserting the church elders information into the database--

INSERT INTO ELDER(ELDER\_ID, FIRST\_NAME, LAST\_NAME, OFFICE\_PHONE\_NUMBER, OFFICE\_EMAIL\_ADDRESS, OFFICE)

VALUES(01, 'JOHN', 'JOHNSON', '7037302227', 'ELDER1@GMAIL.COM', 'E101');

INSERT INTO ELDER(ELDER\_ID, FIRST\_NAME, LAST\_NAME, OFFICE\_PHONE\_NUMBER, OFFICE\_EMAIL\_ADDRESS, OFFICE)

VALUES(02, 'PETER', 'SAMSON', '7037302228', 'ELDER2@GMAIL.COM', 'E102');

INSERT INTO ELDER(ELDER\_ID, FIRST\_NAME, LAST\_NAME, OFFICE\_PHONE\_NUMBER, OFFICE\_EMAIL\_ADDRESS, OFFICE)

VALUES(03, 'JAMES', 'MILLER', '7037302228', 'ELDER3@GMAIL.COM', 'E103');

COMMIT;

UPDATE ELDER SET DEPARTMENT\_ID='100' WHERE ELDER\_ID='1';

UPDATE ELDER SET DEPARTMENT\_ID='200' WHERE ELDER\_ID='1';

UPDATE ELDER SET DEPARTMENT\_ID='300' WHERE ELDER\_ID='2';

UPDATE ELDER SET DEPARTMENT\_ID='400' WHERE ELDER\_ID='2';

UPDATE ELDER SET DEPARTMENT\_ID='500' WHERE ELDER\_ID='3';

--Inserting the church pastor’s information into the database--

INSERT INTO PASTOR(PASTOR\_ID, FIRST\_NAME, LAST\_NAME, OFFICE\_PHONE\_NUMBER, OFFICE\_EMAIL\_ADDRESS, OFFICE)

VALUES('P100', 'PAUL', 'PETERS', '7037302230', 'CHPASTOR@GMAIL.COM', 'P101');

COMMIT;

--Inserting the different positions that can be held into the database--

INSERT INTO POSITION(POSITION\_ID, DIRECTOR, ASSISTANT\_DIRECTOR, SECRETARY, TREASURER)

VALUES('T01', 'CARLTON BANKS', 'PHILIP BANKS', 'VIVIAN BANKS', 'HILLARY BANKS');

INSERT INTO POSITION(POSITION\_ID, DIRECTOR, ASSISTANT\_DIRECTOR, SECRETARY, TREASURER)

VALUES('S01', 'ASHLEY BANKS', 'JEFFREY BUTLER', 'N/A', 'N/A');

INSERT INTO POSITION(POSITION\_ID, DIRECTOR, ASSISTANT\_DIRECTOR, SECRETARY, TREASURER)

VALUES('CPMO1', 'PHILIP BANKS', 'VY SMITH', 'HELEN SMITH', 'VIVIAN BANKS');

INSERT INTO POSITION(POSITION\_ID, DIRECTOR, ASSISTANT\_DIRECTOR, SECRETARY, TREASURER)

VALUES('M01', 'WILL SMITH', 'NICHOLAS BANKS', 'ASHLEY BANKS', 'CARLTON BANKS');

INSERT INTO POSITION(POSITION\_ID, DIRECTOR, ASSISTANT\_DIRECTOR, SECRETARY, TREASURER)

VALUES('Y01', 'JEFFREY BUTLER', 'NICHOLAS BANKS', 'WILL SMITH', 'VY SMITH');

COMMIT;

--Query 1--

SELECT \* FROM CHURCH\_MEMBER;

--Query 2--

SELECT FIRST\_NAME, LAST\_NAME, SEX, EMAIL\_ADDRESS, PHONE\_NUMBER FROM CHURCH\_MEMBER;

--Query 3--

SELECT \* FROM CONFERENCE\_INFO;

--Query 4--

SELECT CHURCH\_MEMBER.MEMBER\_ID, CHURCH\_MEMBER.FIRST\_NAME, CHURCH\_MEMBER.LAST\_NAME, CHURCH\_MEMBER.EMAIL\_ADDRESS,

CHURCH\_MEMBER.SEX, CHURCH\_MEMBER.TERM, CHURCH\_MEMBER.PHONE\_NUMBER, POSITION.POSITION\_ID, POSITION.DIRECTOR,

POSITION.ASSISTANT\_DIRECTOR, POSITION.SECRETARY, POSITION.TREASURER FROM CHURCH\_MEMBER

FULL JOIN POSITION ON CHURCH\_MEMBER.POSITION\_ID=POSITION.POSITION\_ID;

--Query 5--

SELECT \* FROM CHURCH\_MEMBER ORDER BY LAST\_NAME ASC;

--Query 6--

SELECT DEPARTMENT.DEPARTMENT\_ID, DEPARTMENT.DEPARTMENT\_NAME, ELDER.FIRST\_NAME, ELDER.LAST\_NAME, CONFERENCE\_LEADER.CONFERENCE\_LEADER\_ID

FROM DEPARTMENT RIGHT JOIN ELDER ON DEPARTMENT.ELDER\_ID=ELDER.ELDER\_ID

LEFT JOIN CONFERENCE\_LEADER ON DEPARTMENT.CONFERENCE\_LEADER\_ID=CONFERENCE.CONFERENCE\_LEADER\_ID

FETCH FIRST 10 ROWS ONLY;

--Query 7--

SELECT DISTINCT ELDER.ELDER\_ID, ELDER.FIRST\_NAME, ELDER.LAST\_NAME, DEPARTMENT.DEPARTMENT\_ID, DEPARTMENT.DEPARTMENT\_NAME, CHURCH\_MEMBER.MEMBER\_ID FROM ELDER

RIGHT JOIN DEPARTMENT ON ELDER.DEPARTMENT\_ID=DEPARTMENT.DEPARTMENT\_ID

LEFT JOIN CHURCH\_MEMBER ON DEPARTMENT.MEMBER\_ID=CHURCH\_MEMBER.MEMBER\_ID;

--Query 8--

SELECT LAST\_NAME, FIRST\_NAME, SEX, COUNT(\*) FROM CHURCH\_MEMBER GROUP BY LAST\_NAME, FIRST\_NAME, SEX;

--Query 9--

SELECT \* FROM ELDER WHERE ELDER\_ID IN (SELECT DEPARTMENT.ELDER\_ID FROM DEPARTMENT);

--Query 10--

SELECT OFFICE\_EMAIL\_ADDRESS, LENGTH(OFFICE\_EMAIL\_ADDRESS) FROM PASTOR;

--Query 11--

DELETE FROM POSITION WHERE DIRECTOR=’PHILIP’;

ROLLBACK;

--Query 12—

UPDATE ELDER SET ELDER.FIRST\_NAME=’LUKE’ WHERE ELDER.FIRST\_NAME=’JOHN’;

SELECT \* FROM ELDER;

ROLLBACK;

SELECT \* FROM ELDER;

--Query 13--

SELECT \* FROM DEPARTMENT WHERE ELDER\_ID=(SELECT ELDER\_ID FROM ELDER WHERE ELDER\_ID='1');

--Query 14--

SELECT \* FROM CONFERENCE\_LEADER WHERE PASTOR\_ID=(SELECT PASTOR\_ID FROM PASTOR);

--Query 15--

SELECT ELDER\_ID, MEMBER\_ID, DEPARTMENT\_NAME, DEPARTMENT\_FUNCTION FROM DEPARTMENT ORDER BY DEPARTMENT\_NAME DESC;

--Query 16--

SELECT CHURCH\_MEMBER.LAST\_NAME, CHURCH\_MEMBER.FIRST\_NAME, (SELECT COUNT(\*) FROM CHURCH\_MEMBER WHERE LAST\_NAME='BANKS') AS FAMILY

FROM CHURCH\_MEMBER WHERE LAST\_NAME='BANKS' GROUP BY CHURCH\_MEMBER.LAST\_NAME, CHURCH\_MEMBER.FIRST\_NAME ORDER BY FIRST\_NAME ASC;

--Query 17--

SELECT CHURCH\_MEMBER.FIRST\_NAME, CHURCH\_MEMBER.LAST\_NAME, POSITION.POSITION\_ID FROM CHURCH\_MEMBER

INNER JOIN POSITION ON CHURCH\_MEMBER.POSITION\_ID=POSITION.POSITION\_ID

WHERE POSITION.POSITION\_ID='CPMO1' ORDER BY CHURCH\_MEMBER.LAST\_NAME;

--Query 18--

SELECT CHURCH\_MEMBER.FIRST\_NAME, CHURCH\_MEMBER.LAST\_NAME, POSITION.POSITION\_ID, DEPARTMENT.DEPARTMENT\_ID,

CONFERENCE\_LEADER.CONFERENCE\_LEADER\_ID FROM CHURCH\_MEMBER

INNER JOIN POSITION ON CHURCH\_MEMBER.POSITION\_ID=POSITION.POSITION\_ID

INNER JOIN DEPARTMENT ON CHURCH\_MEMBER.MEMBER\_ID=DEPARTMENT.MEMBER\_ID

INNER JOIN CONFERENCE\_LEADER ON DEPARTMENT.CONFERENCE\_LEADER\_ID=CONFERENCE\_LEADER.CONFERENCE\_LEADER\_ID

ORDER BY CHURCH\_MEMBER.LAST\_NAME ASC;

--Query 19--

SELECT SEX, COUNT(CASE WHEN SEX='M' THEN 1 END) AS MALE, COUNT(CASE WHEN SEX='F' THEN 1 END) AS FEMALE FROM CHURCH\_MEMBER

GROUP BY SEX;

--Query 20--

SELECT ELDER.FIRST\_NAME, ELDER.LAST\_NAME, ELDER.OFFICE\_PHONE\_NUMBER, DEPARTMENT.DEPARTMENT\_NAME, CONFERENCE\_LEADER.FIRST\_NAME,

CONFERENCE\_LEADER.LAST\_NAME, CONFERENCE\_LEADER.OFFICE\_EMAIL\_ADDRESS FROM ELDER

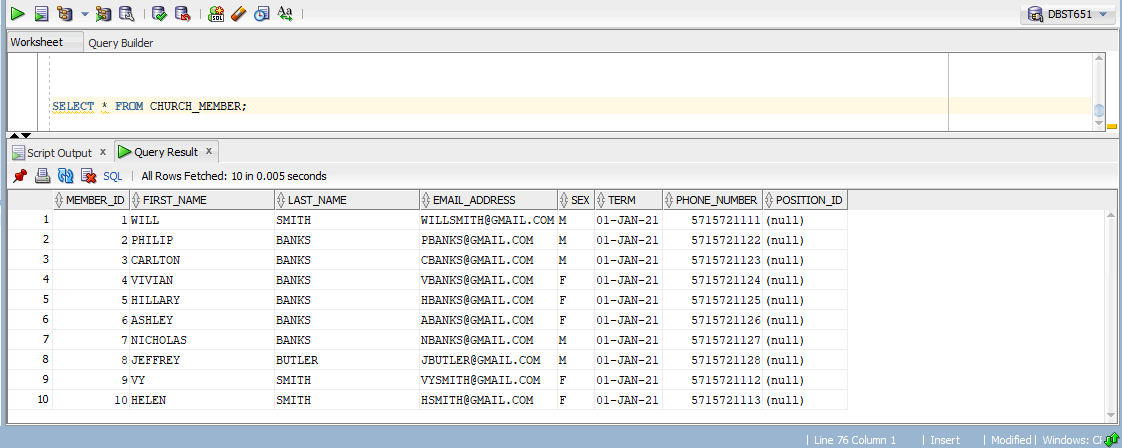
INNER JOIN DEPARTMENT ON ELDER.DEPARTMENT\_ID=DEPARTMENT.DEPARTMENT\_ID

INNER JOIN CONFERENCE\_LEADER ON DEPARTMENT.CONFERENCE\_LEADER\_ID=CONFERENCE\_LEADER.CONFERENCE\_LEADER\_ID

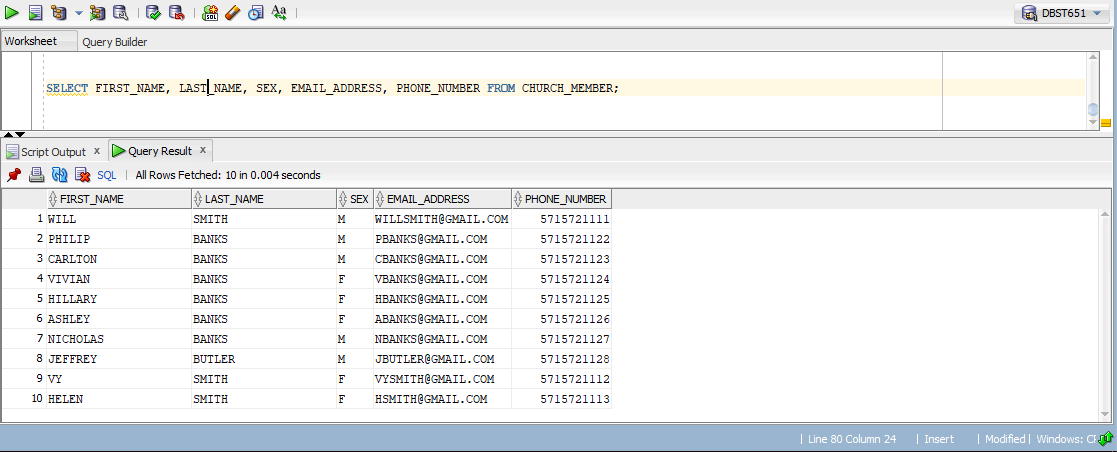
ORDER BY ELDER.LAST\_NAME ASC;

**Query Output**

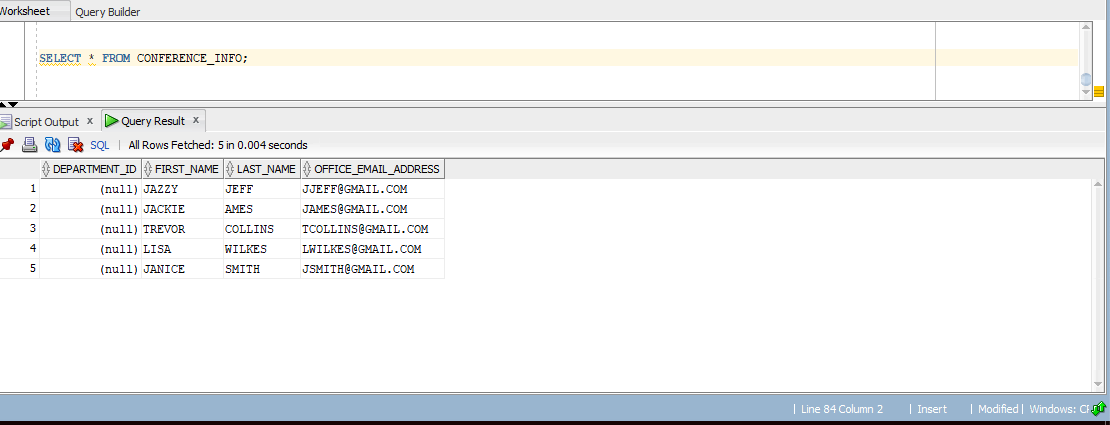
1.



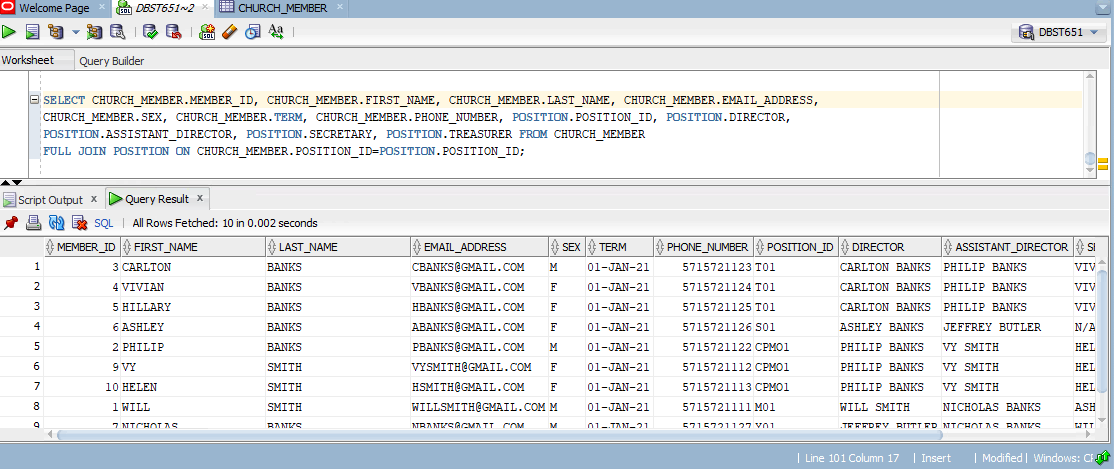
2.



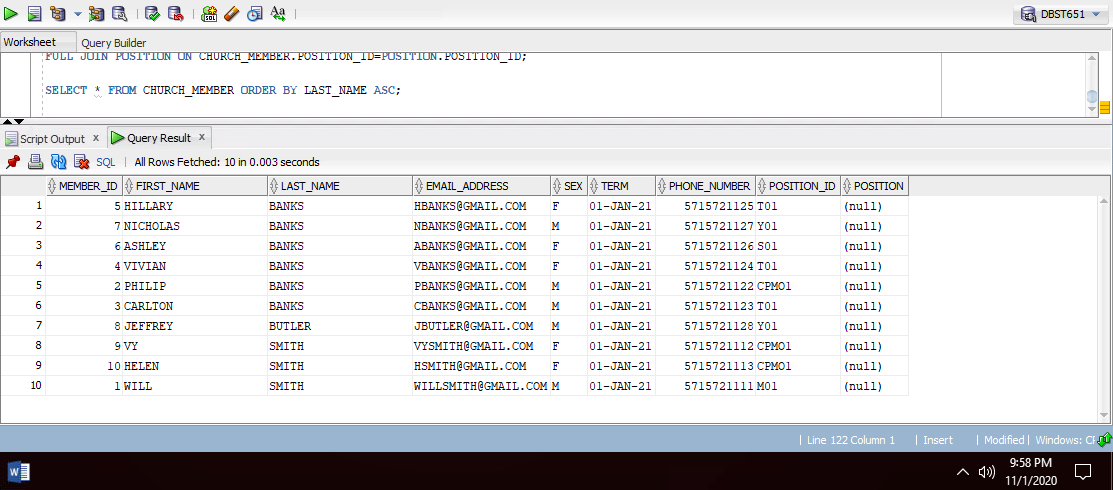
3.



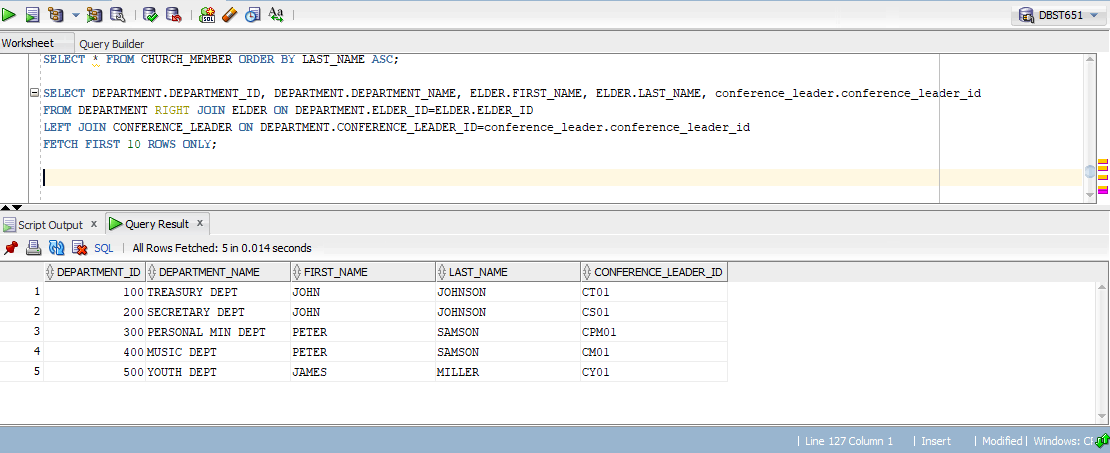
4.



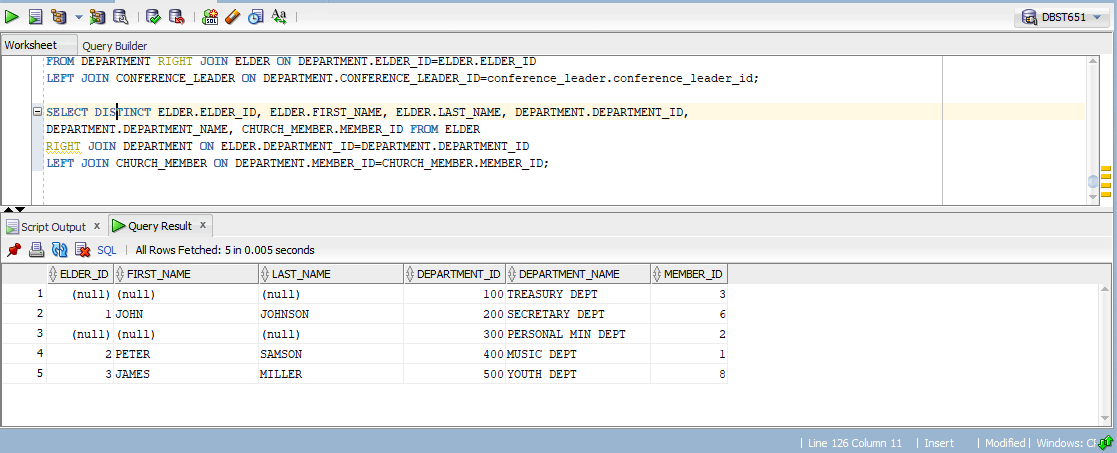
5.



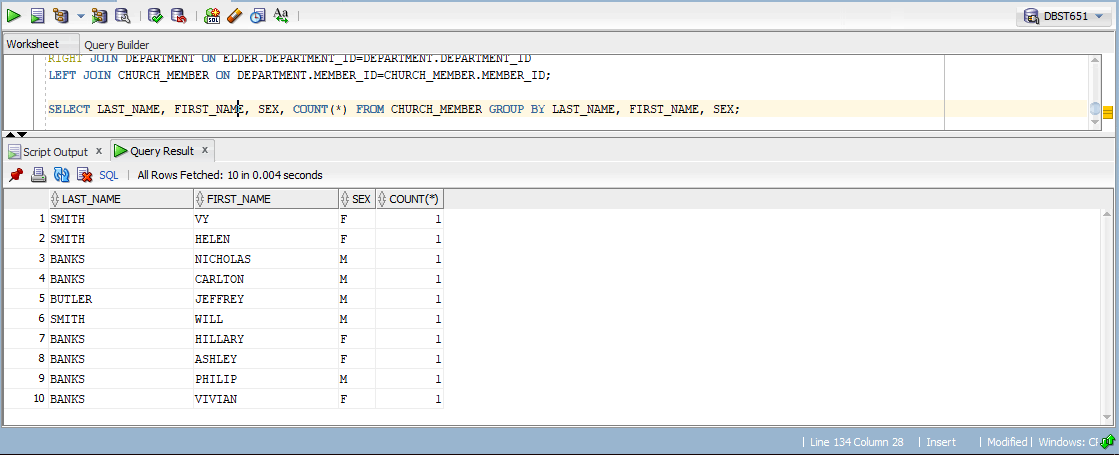
6.

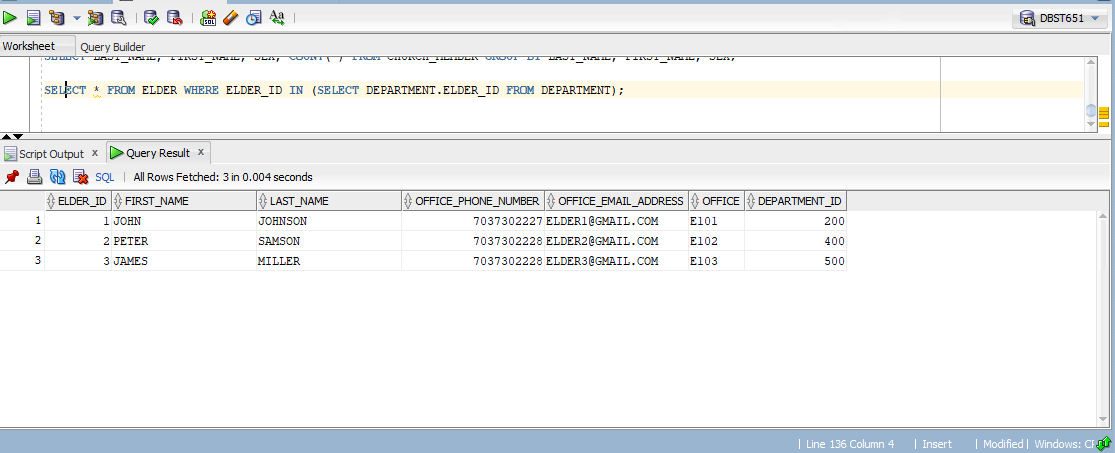


7.

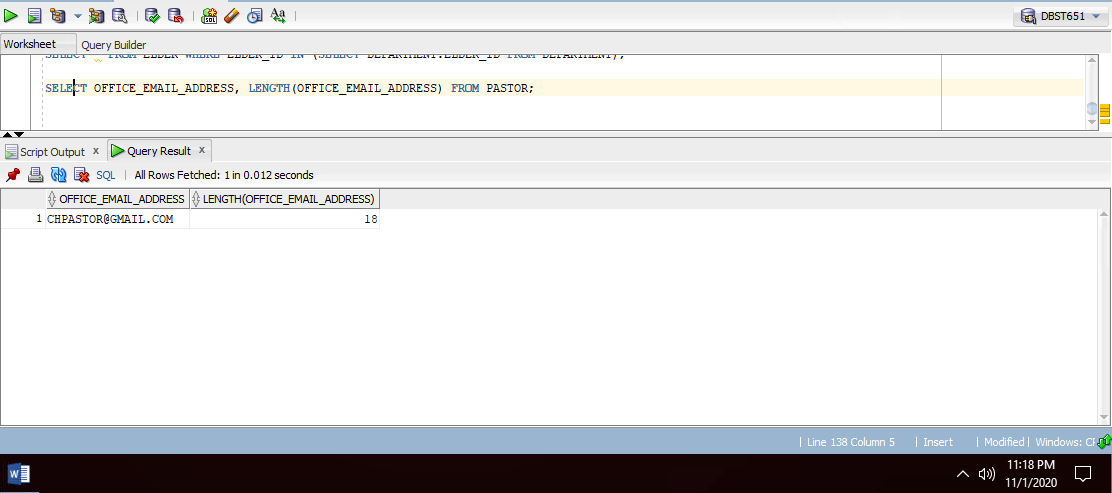


8.

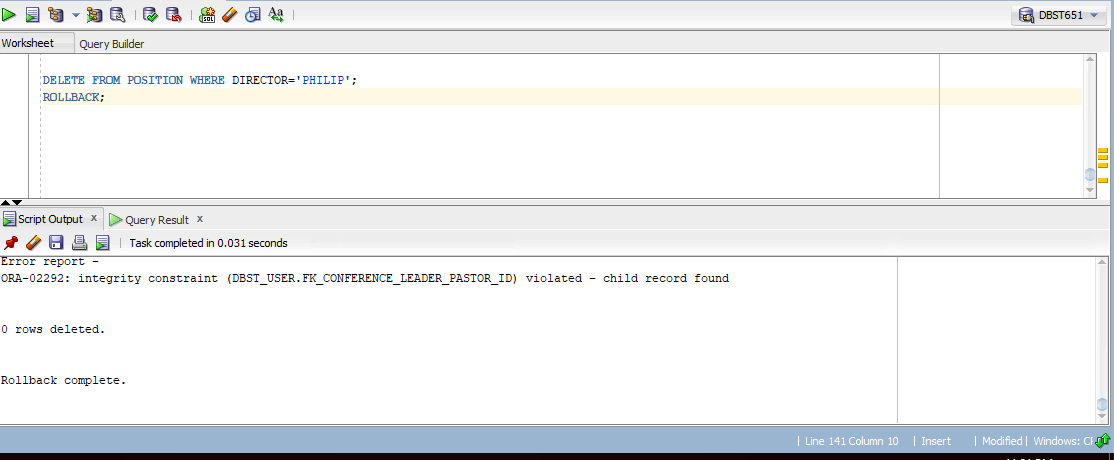


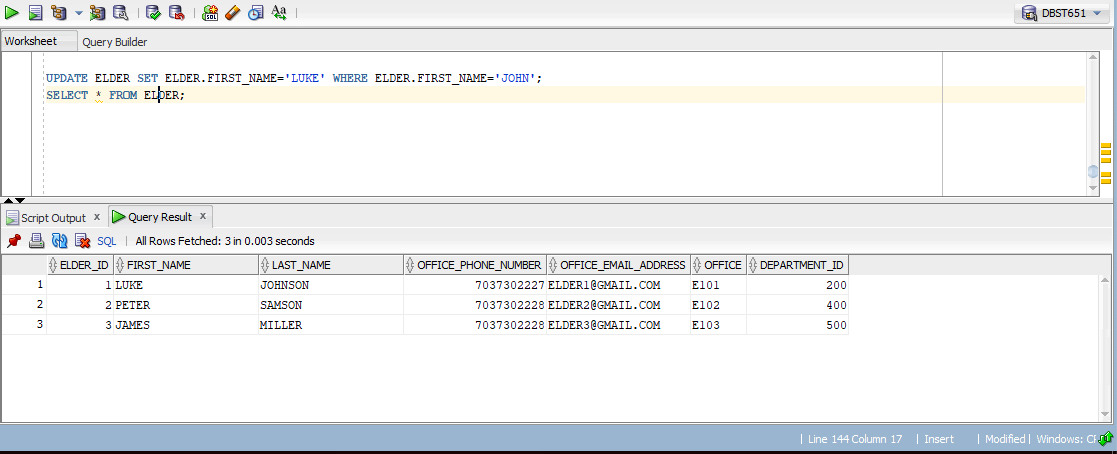
9. 

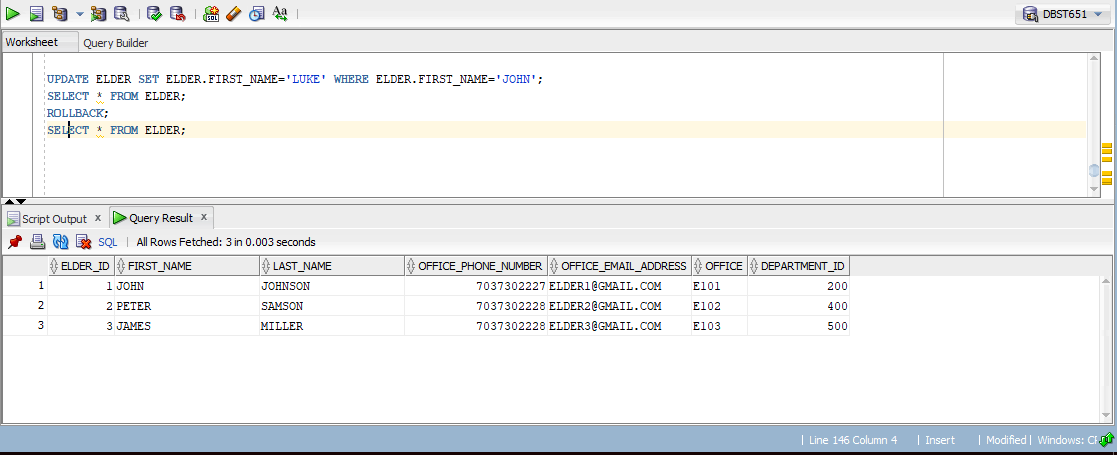
10.



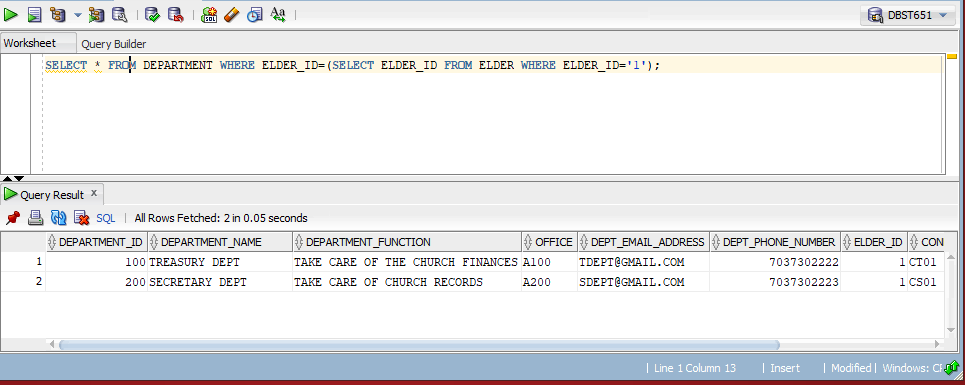
11.



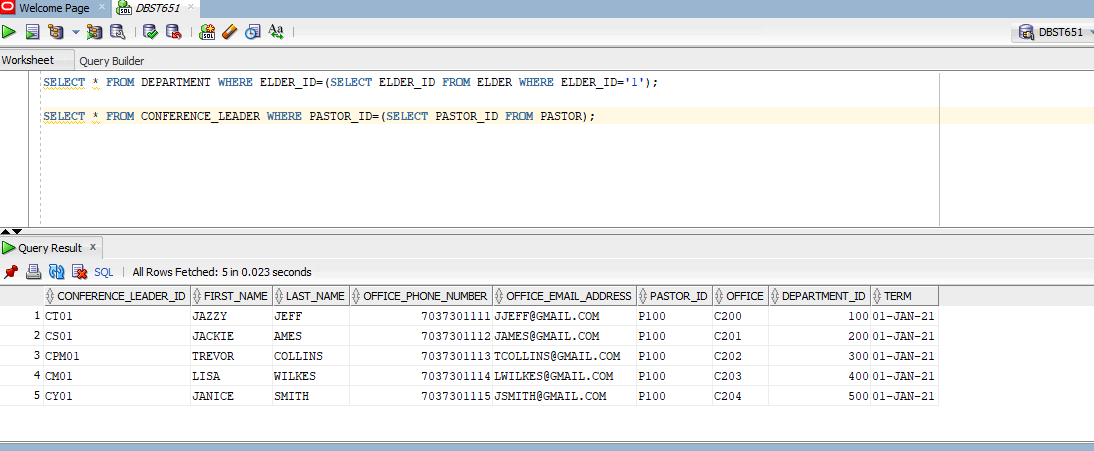
12. 



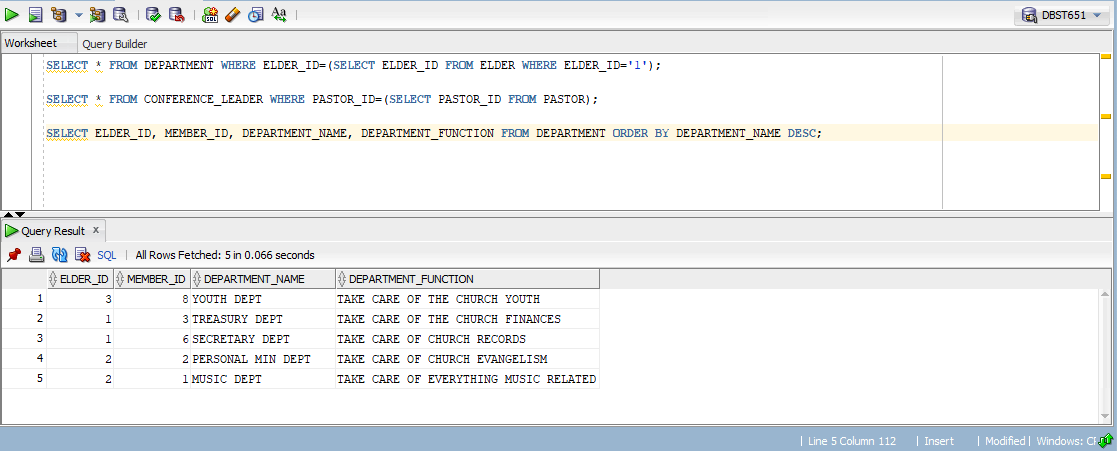
13.



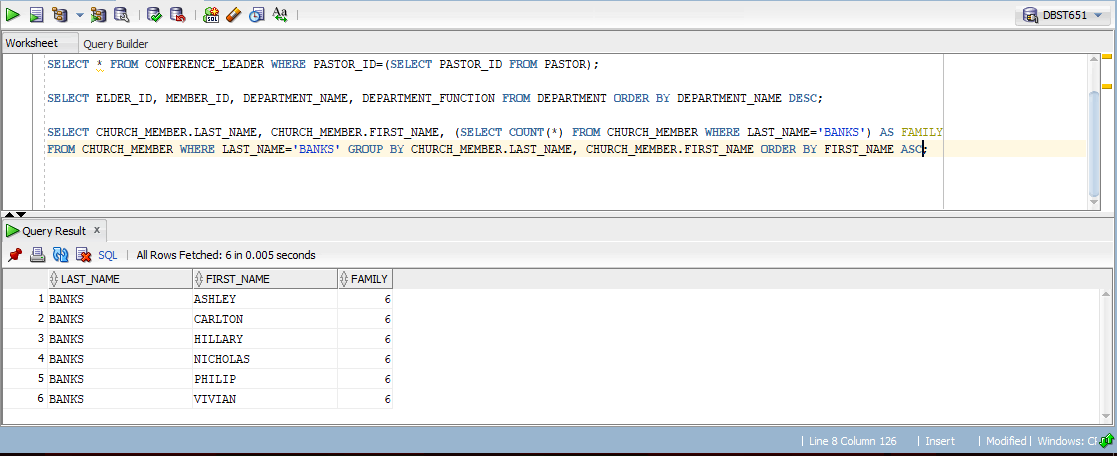
14.



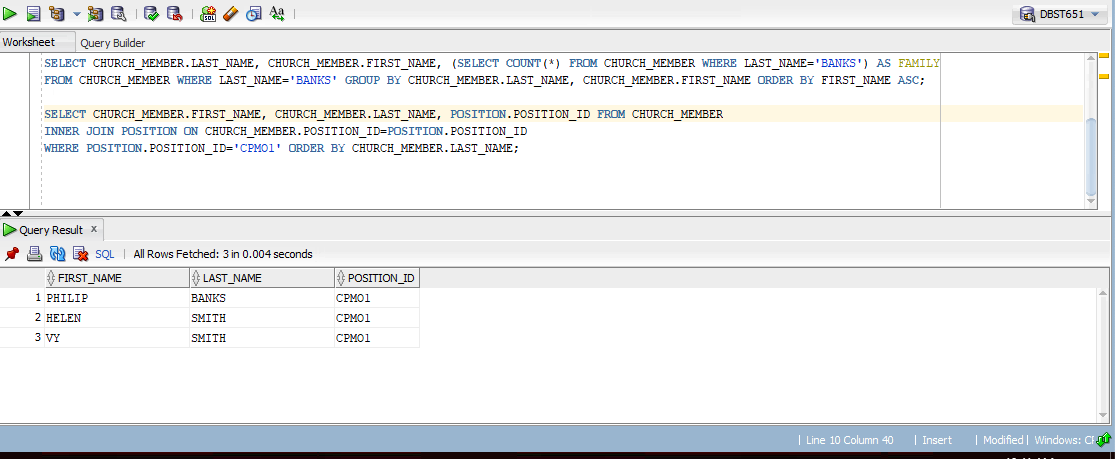
15.



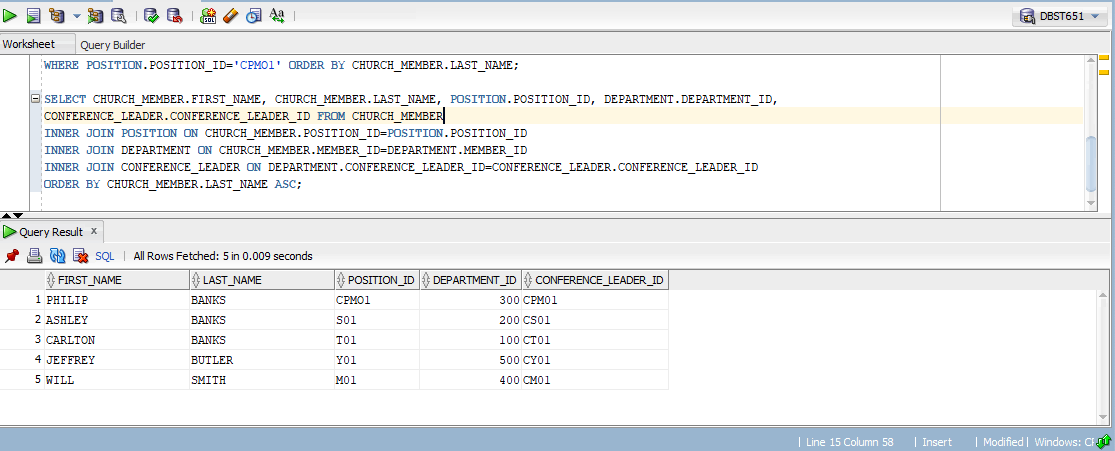
16.



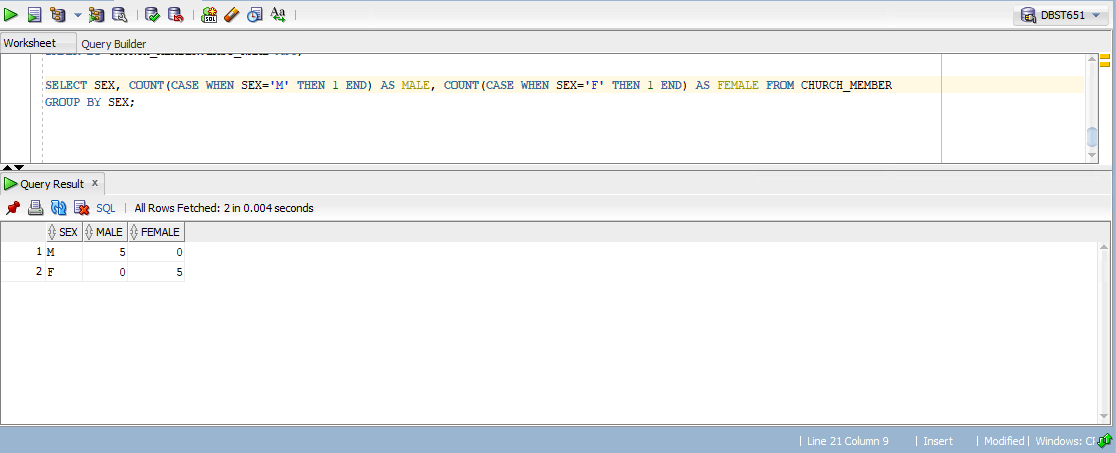
17.



18.



19.



20.

