SQL Databases Final Project

Sankalp Mundra

Table of contents

01

Introduction

Who? What? Why? How?

02

Database Schema Diagram (ERD)

Describing the database structure through an entity-relationship diagram constructed using the Crow's Foot notation

Database Schema for Northeastern University Registry

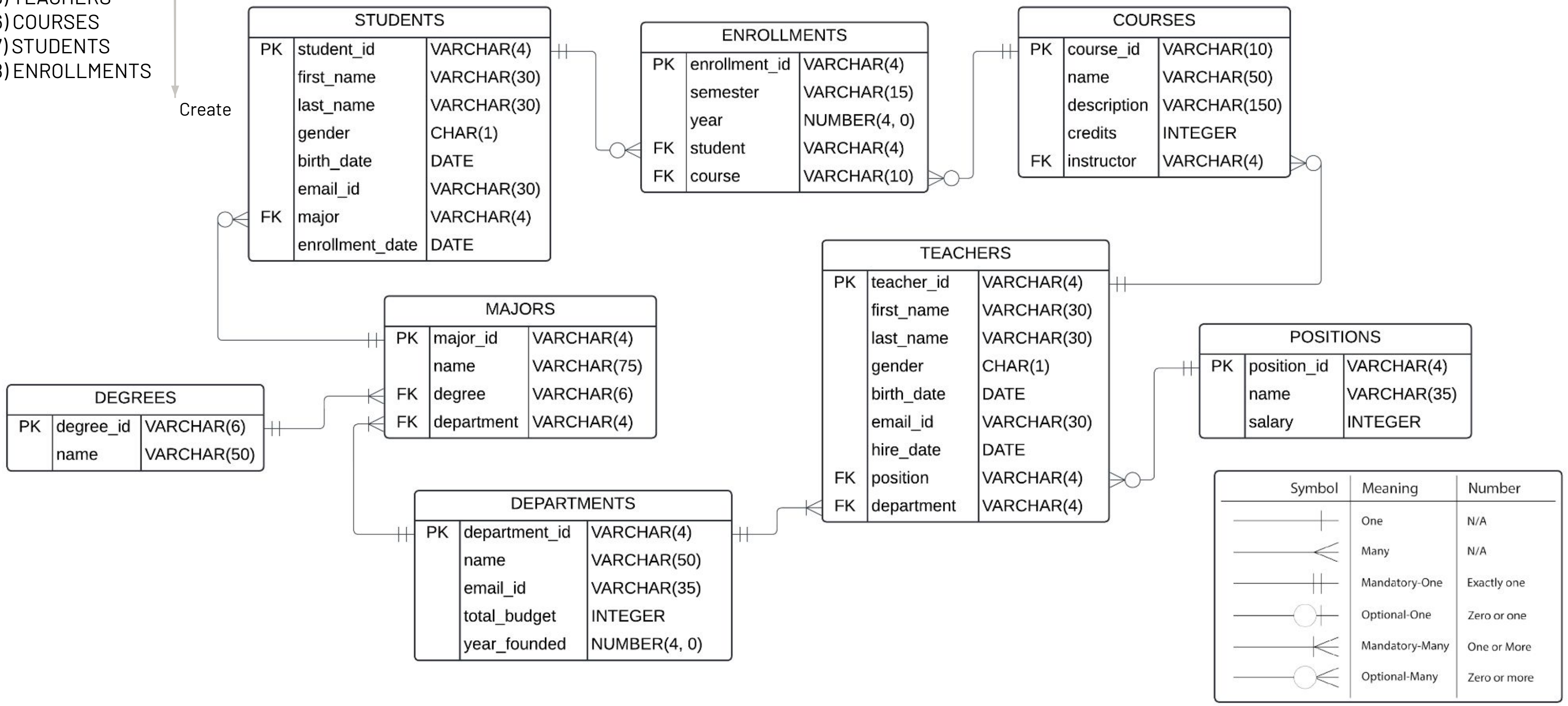
ERD Diagram By Sankalp Mundra

Table Hierarchy:

1) DEGREES
2) POSITIONS
3) DEPARTMENTS
4) MAJORS
5) TEACHERS
6) COURSES
7) STUDENTS
8) ENROLLMENTS

Delete

Create



STUDENTS

PK	student_id	VARCHAR(4)
	first_name	VARCHAR(30)
	last_name	VARCHAR(30)
	gender	CHAR(1)
	birth_date	DATE
	email_id	VARCHAR(30)
FK	major	VARCHAR(4)
	enrollment_date	DATE

ENROLLMENTS

PK	enrollment_id	VARCHAR(4)
	semester	VARCHAR(15)
	year	NUMBER(4, 0)
FK	student	VARCHAR(4)
FK	course	VARCHAR(10)

COURSES

PK	course_id	VARCHAR(10)
	name	VARCHAR(50)
	description	VARCHAR(150)
	credits	INTEGER
FK	instructor	VARCHAR(4)

TEACHERS

PK	teacher_id	VARCHAR(4)
	first_name	VARCHAR(30)
	last_name	VARCHAR(30)
	gender	CHAR(1)
	birth_date	DATE
	email_id	VARCHAR(30)
	hire_date	DATE
FK	position	VARCHAR(4)
FK	department	VARCHAR(4)

MAJORS

PK	major_id	VARCHAR(4)
	name	VARCHAR(75)
FK	degree	VARCHAR(6)
FK	department	VARCHAR(4)

DEGREES

PK	degree_id	VARCHAR(6)
	name	VARCHAR(50)

DEPARTMENTS

PK	department_id	VARCHAR(4)
	name	VARCHAR(50)
	email_id	VARCHAR(35)
	total_budget	INTEGER
	year_founded	NUMBER(4, 0)

POSITIONS

PK	position_id	VARCHAR(4)
	name	VARCHAR(35)
	salary	INTEGER

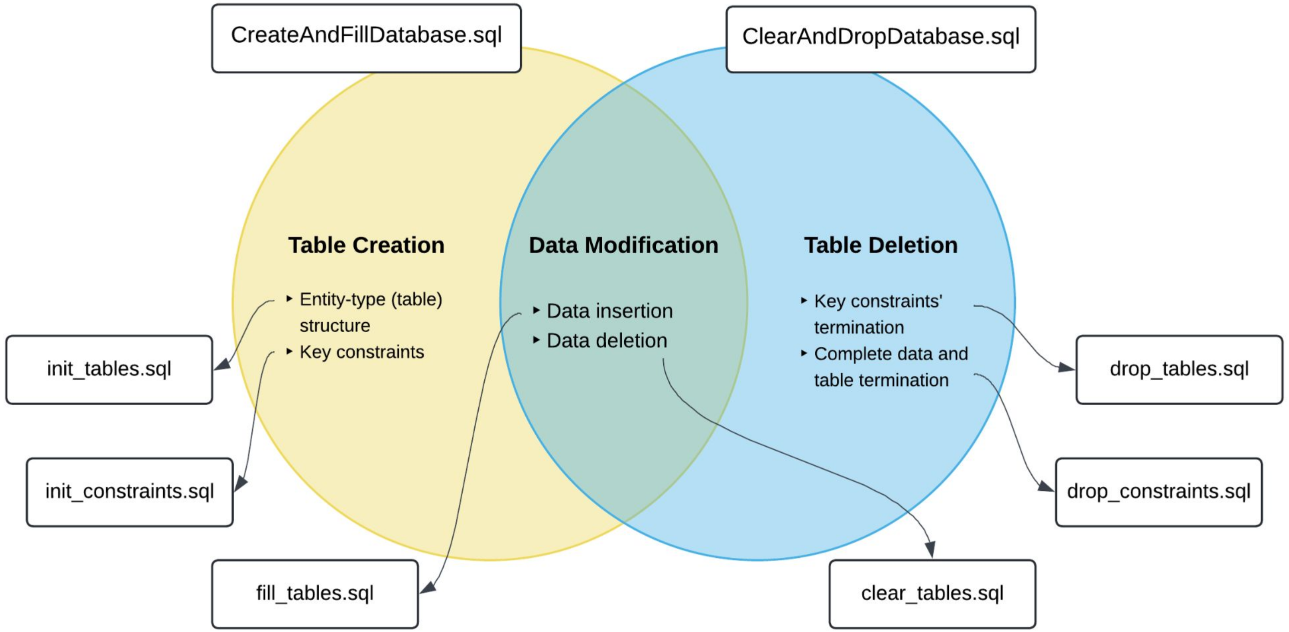
Symbol	Meaning	Number
	One	N/A
	Many	N/A
	Mandatory-One	Exactly one
	Optional-One	Zero or one
	Mandatory-Many	One or More
	Optional-Many	Zero or more

03

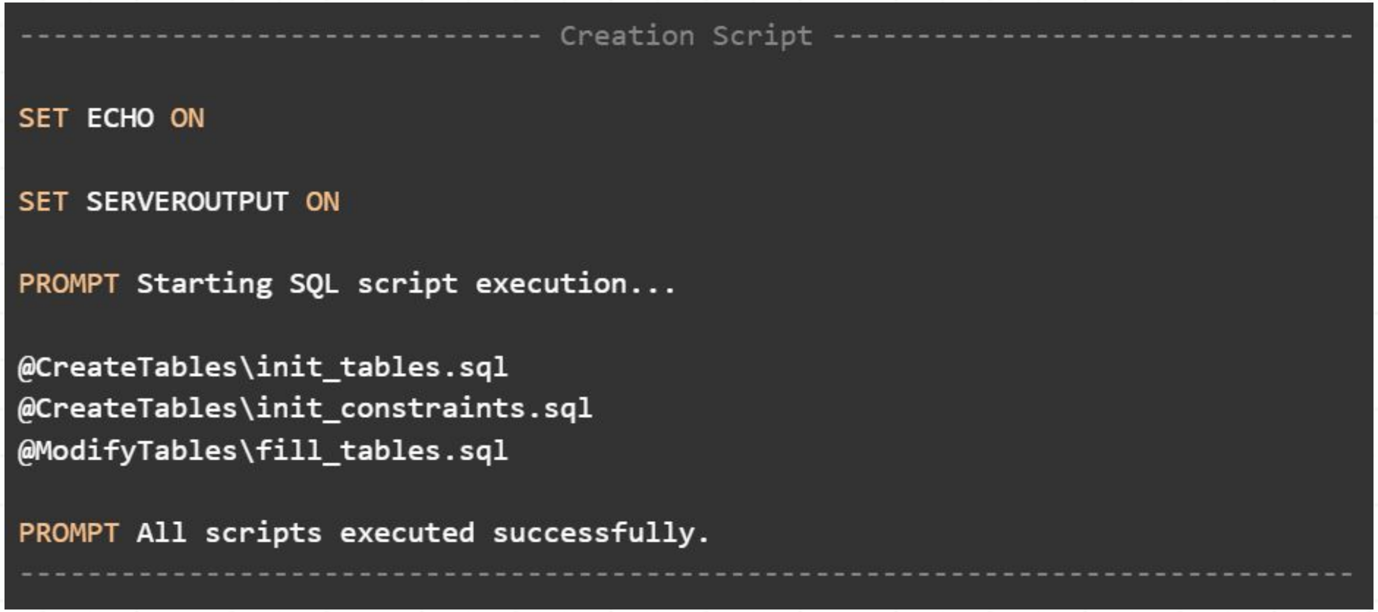
Database Scripts

Table Creation ⟶ Data Modification ⟶ Table Deletion

Scripts Venn Diagram

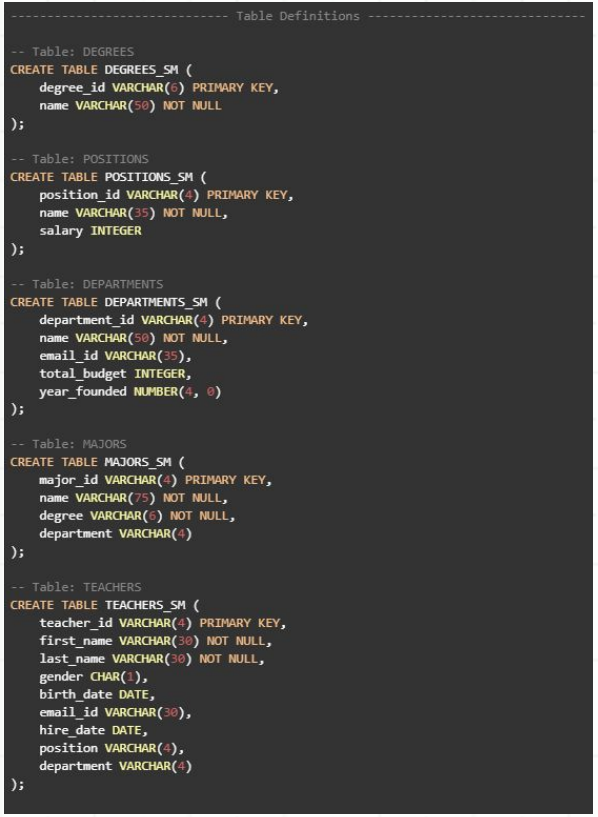


CreateAndFillDatabase.sql

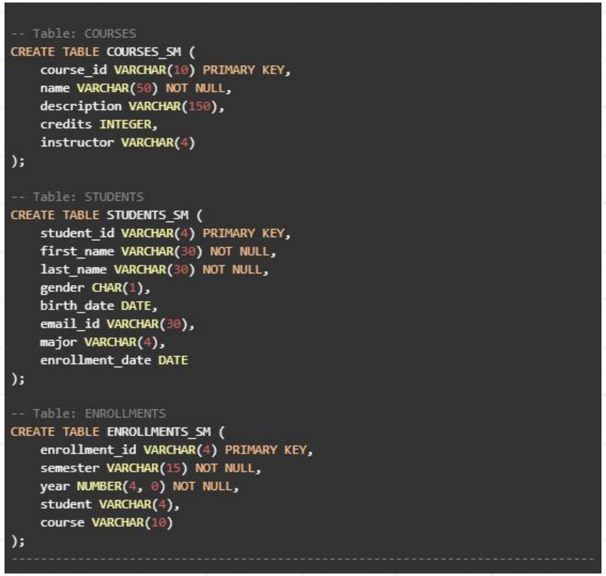


```
------------------------------ Creation Script ------------------------------

SET ECHO ON

SET SERVEROUTPUT ON

PROMPT Starting SQL script execution...

@CreateTables\init_tables.sql
@CreateTables\init_constraints.sql
@ModifyTables\fill_tables.sql

PROMPT All scripts executed successfully.
-----------------------------------------------------------------------------
```

init_tables.sql

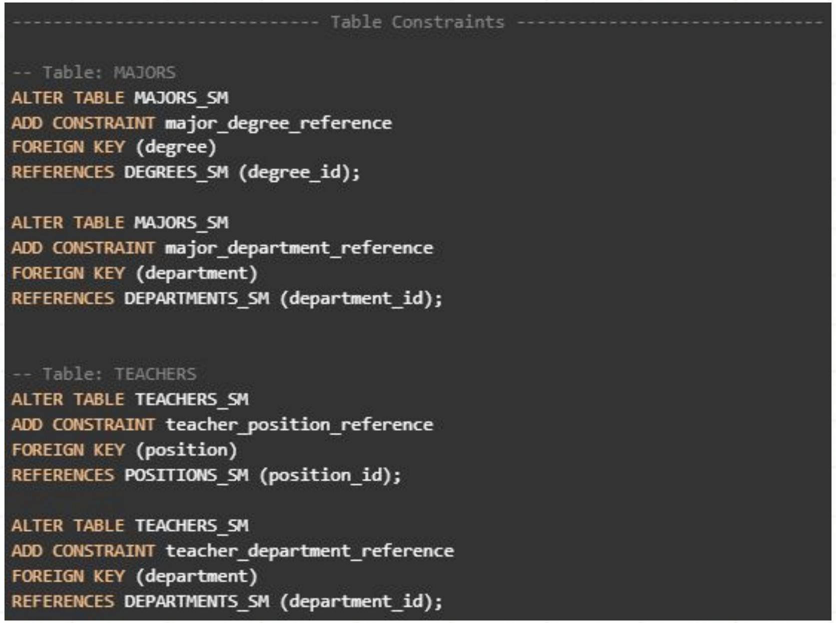


```sql
------------------------------ Table Definitions ------------------------------

-- Table: DEGREES
CREATE TABLE DEGREES_SM (
    degree_id VARCHAR(6) PRIMARY KEY,
    name VARCHAR(50) NOT NULL
);

-- Table: POSITIONS
CREATE TABLE POSITIONS_SM (
    position_id VARCHAR(4) PRIMARY KEY,
    name VARCHAR(35) NOT NULL,
    salary INTEGER
);

-- Table: DEPARTMENTS
CREATE TABLE DEPARTMENTS_SM (
    department_id VARCHAR(4) PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    email_id VARCHAR(35),
    total_budget INTEGER,
    year_founded NUMBER(4, 0)
);

-- Table: MAJORS
CREATE TABLE MAJORS_SM (
    major_id VARCHAR(4) PRIMARY KEY,
    name VARCHAR(75) NOT NULL,
    degree VARCHAR(6) NOT NULL,
    department VARCHAR(4)
);

-- Table: TEACHERS
CREATE TABLE TEACHERS_SM (
    teacher_id VARCHAR(4) PRIMARY KEY,
    first_name VARCHAR(30) NOT NULL,
    last_name VARCHAR(30) NOT NULL,
    gender CHAR(1),
    birth_date DATE,
    email_id VARCHAR(30),
    hire_date DATE,
    position VARCHAR(4),
    department VARCHAR(4)
);
```

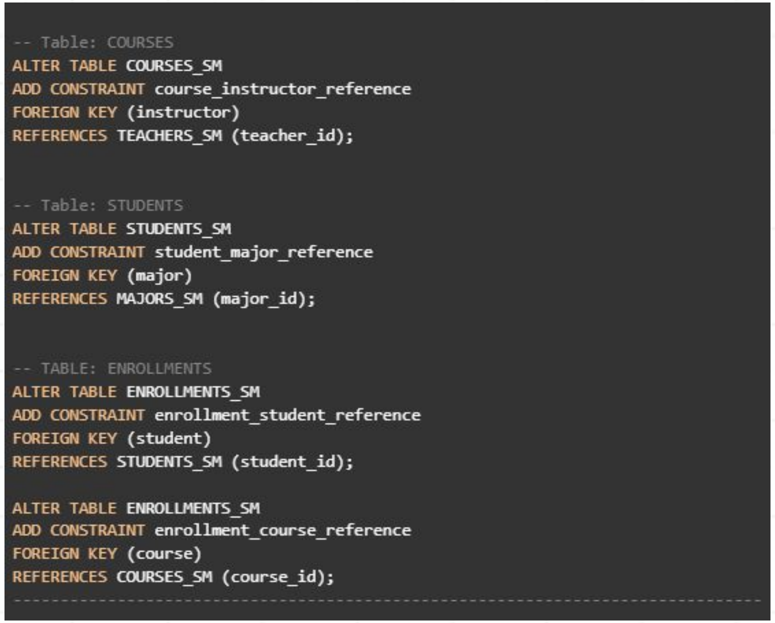


```sql
-- Table: COURSES
CREATE TABLE COURSES_SM (
    course_id VARCHAR(10) PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    description VARCHAR(150),
    credits INTEGER,
    instructor VARCHAR(4)
);

-- Table: STUDENTS
CREATE TABLE STUDENTS_SM (
    student_id VARCHAR(4) PRIMARY KEY,
    first_name VARCHAR(30) NOT NULL,
    last_name VARCHAR(30) NOT NULL,
    gender CHAR(1),
    birth_date DATE,
    email_id VARCHAR(30),
    major VARCHAR(4),
    enrollment_date DATE
);

-- Table: ENROLLMENTS
CREATE TABLE ENROLLMENTS_SM (
    enrollment_id VARCHAR(4) PRIMARY KEY,
    semester VARCHAR(15) NOT NULL,
    year NUMBER(4, 0) NOT NULL,
    student VARCHAR(4),
    course VARCHAR(10)
);
--------------------------------------------------------------------------------
```

init_constraints.sql

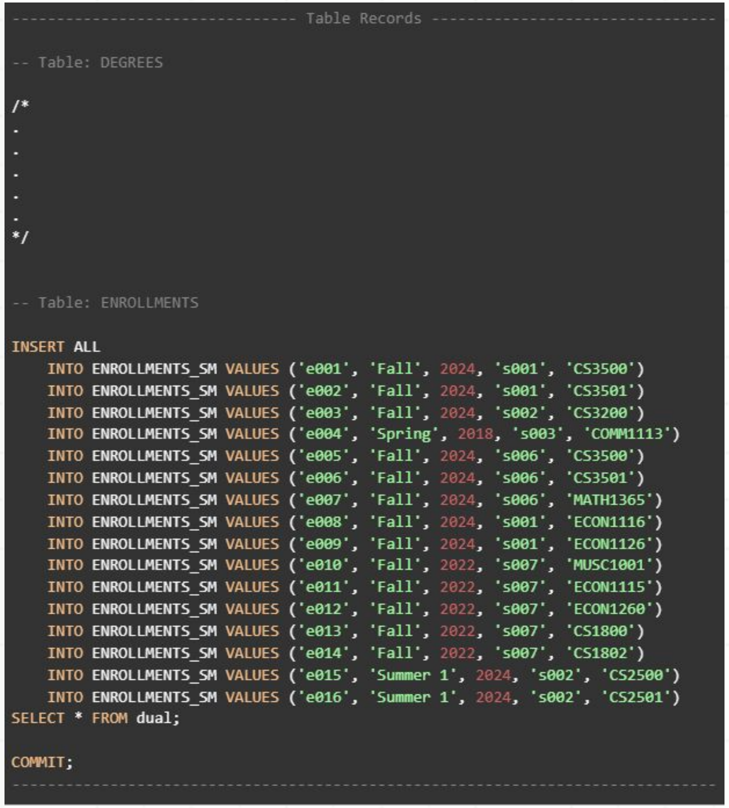


```sql
------------------------------ Table Constraints ------------------------------

-- Table: MAJORS
ALTER TABLE MAJORS_SM
ADD CONSTRAINT major_degree_reference
FOREIGN KEY (degree)
REFERENCES DEGREES_SM (degree_id);

ALTER TABLE MAJORS_SM
ADD CONSTRAINT major_department_reference
FOREIGN KEY (department)
REFERENCES DEPARTMENTS_SM (department_id);


-- Table: TEACHERS
ALTER TABLE TEACHERS_SM
ADD CONSTRAINT teacher_position_reference
FOREIGN KEY (position)
REFERENCES POSITIONS_SM (position_id);

ALTER TABLE TEACHERS_SM
ADD CONSTRAINT teacher_department_reference
FOREIGN KEY (department)
REFERENCES DEPARTMENTS_SM (department_id);
```



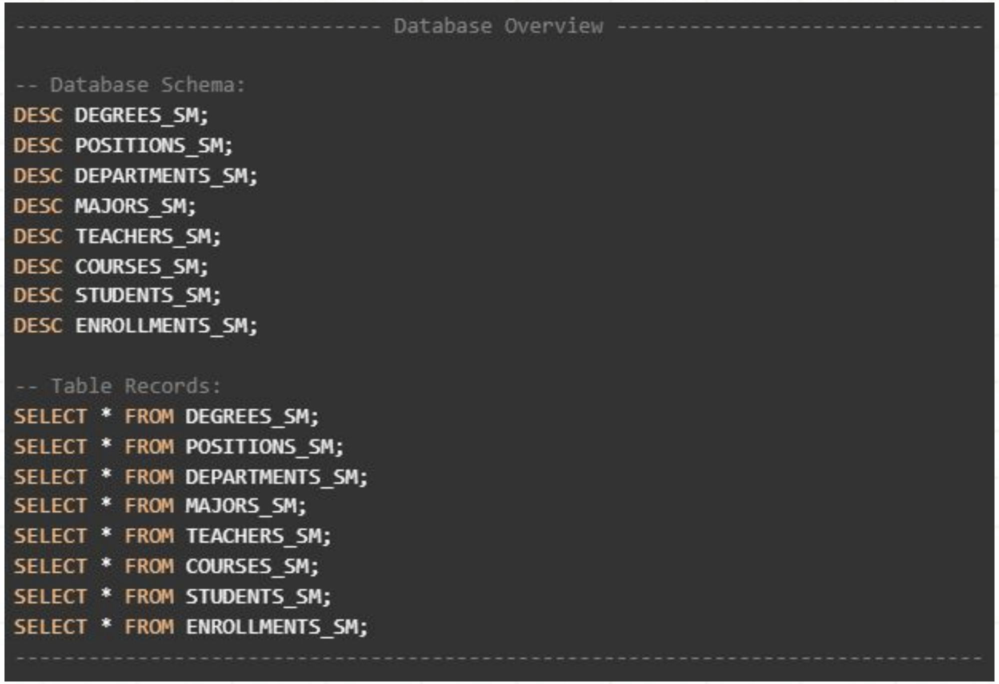
```sql
-- Table: COURSES
ALTER TABLE COURSES_SM
ADD CONSTRAINT course_instructor_reference
FOREIGN KEY (instructor)
REFERENCES TEACHERS_SM (teacher_id);


-- Table: STUDENTS
ALTER TABLE STUDENTS_SM
ADD CONSTRAINT student_major_reference
FOREIGN KEY (major)
REFERENCES MAJORS_SM (major_id);


-- TABLE: ENROLLMENTS
ALTER TABLE ENROLLMENTS_SM
ADD CONSTRAINT enrollment_student_reference
FOREIGN KEY (student)
REFERENCES STUDENTS_SM (student_id);

ALTER TABLE ENROLLMENTS_SM
ADD CONSTRAINT enrollment_course_reference
FOREIGN KEY (course)
REFERENCES COURSES_SM (course_id);
-------------------------------------------------------------------------------
```

fill_tables.sql



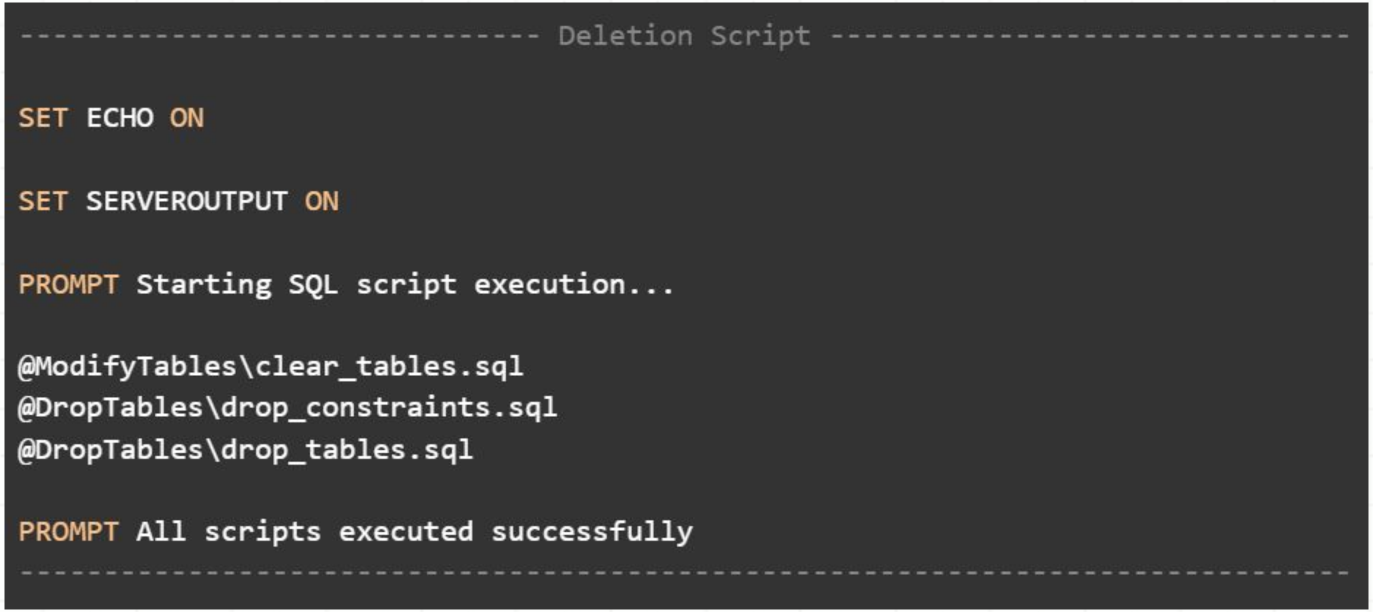
```sql
------------------------------- Table Records -------------------------------

-- Table: DEGREES

/*
.
.
.
.
.
*/


-- Table: ENROLLMENTS

INSERT ALL
    INTO ENROLLMENTS_SM VALUES ('e001', 'Fall', 2024, 's001', 'CS3500')
    INTO ENROLLMENTS_SM VALUES ('e002', 'Fall', 2024, 's001', 'CS3501')
    INTO ENROLLMENTS_SM VALUES ('e003', 'Fall', 2024, 's002', 'CS3200')
    INTO ENROLLMENTS_SM VALUES ('e004', 'Spring', 2018, 's003', 'COMM1113')
    INTO ENROLLMENTS_SM VALUES ('e005', 'Fall', 2024, 's006', 'CS3500')
    INTO ENROLLMENTS_SM VALUES ('e006', 'Fall', 2024, 's006', 'CS3501')
    INTO ENROLLMENTS_SM VALUES ('e007', 'Fall', 2024, 's006', 'MATH1365')
    INTO ENROLLMENTS_SM VALUES ('e008', 'Fall', 2024, 's001', 'ECON1116')
    INTO ENROLLMENTS_SM VALUES ('e009', 'Fall', 2024, 's001', 'ECON1126')
    INTO ENROLLMENTS_SM VALUES ('e010', 'Fall', 2022, 's007', 'MUSC1001')
    INTO ENROLLMENTS_SM VALUES ('e011', 'Fall', 2022, 's007', 'ECON1115')
    INTO ENROLLMENTS_SM VALUES ('e012', 'Fall', 2022, 's007', 'ECON1260')
    INTO ENROLLMENTS_SM VALUES ('e013', 'Fall', 2022, 's007', 'CS1800')
    INTO ENROLLMENTS_SM VALUES ('e014', 'Fall', 2022, 's007', 'CS1802')
    INTO ENROLLMENTS_SM VALUES ('e015', 'Summer 1', 2024, 's002', 'CS2500')
    INTO ENROLLMENTS_SM VALUES ('e016', 'Summer 1', 2024, 's002', 'CS2501')
SELECT * FROM dual;

COMMIT;
-----------------------------------------------------------------------------
```

shortcuts.sql



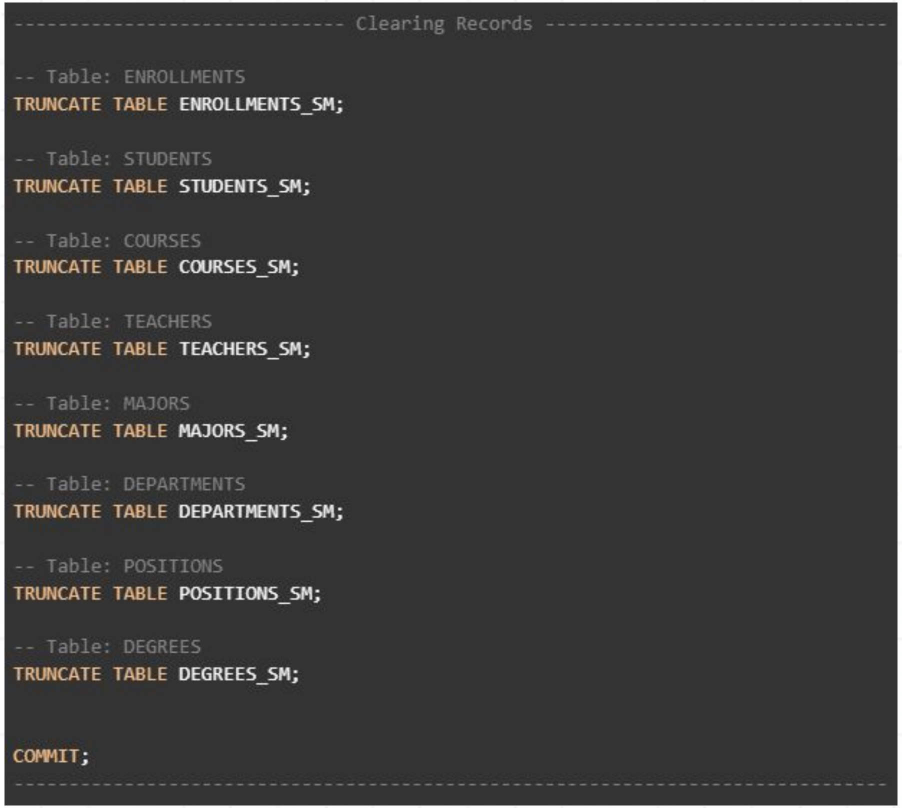
```
----------------------------- Database Overview -----------------------------

-- Database Schema:
DESC DEGREES_SM;
DESC POSITIONS_SM;
DESC DEPARTMENTS_SM;
DESC MAJORS_SM;
DESC TEACHERS_SM;
DESC COURSES_SM;
DESC STUDENTS_SM;
DESC ENROLLMENTS_SM;

-- Table Records:
SELECT * FROM DEGREES_SM;
SELECT * FROM POSITIONS_SM;
SELECT * FROM DEPARTMENTS_SM;
SELECT * FROM MAJORS_SM;
SELECT * FROM TEACHERS_SM;
SELECT * FROM COURSES_SM;
SELECT * FROM STUDENTS_SM;
SELECT * FROM ENROLLMENTS_SM;
------------------------------------------------------------------------------
```

ClearAndDropDatabase.sql



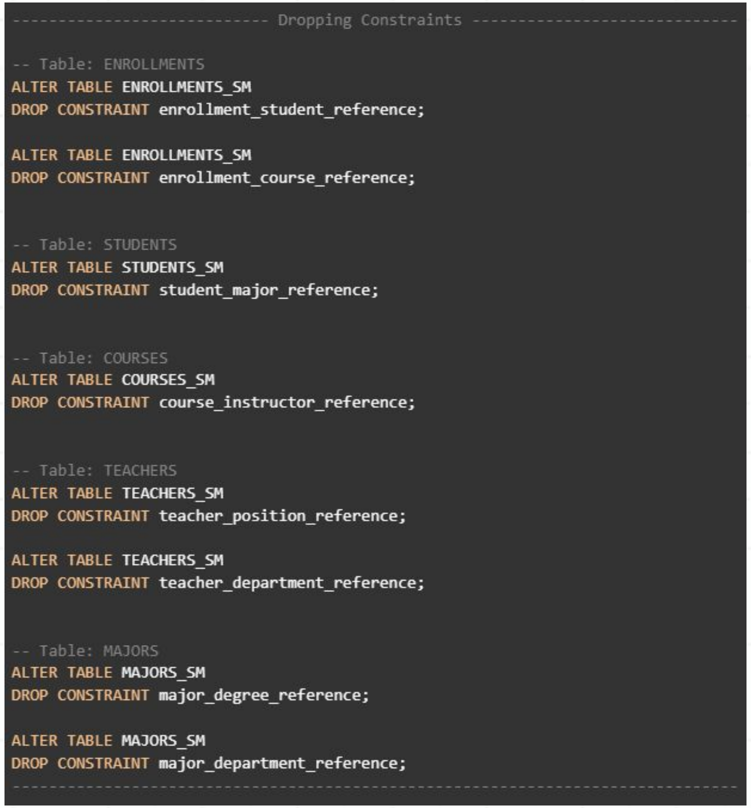
```
------------------------------ Deletion Script ------------------------------

SET ECHO ON

SET SERVEROUTPUT ON

PROMPT Starting SQL script execution...

@ModifyTables\clear_tables.sql
@DropTables\drop_constraints.sql
@DropTables\drop_tables.sql

PROMPT All scripts executed successfully
-----------------------------------------------------------------------------
```

clear_tables.sql



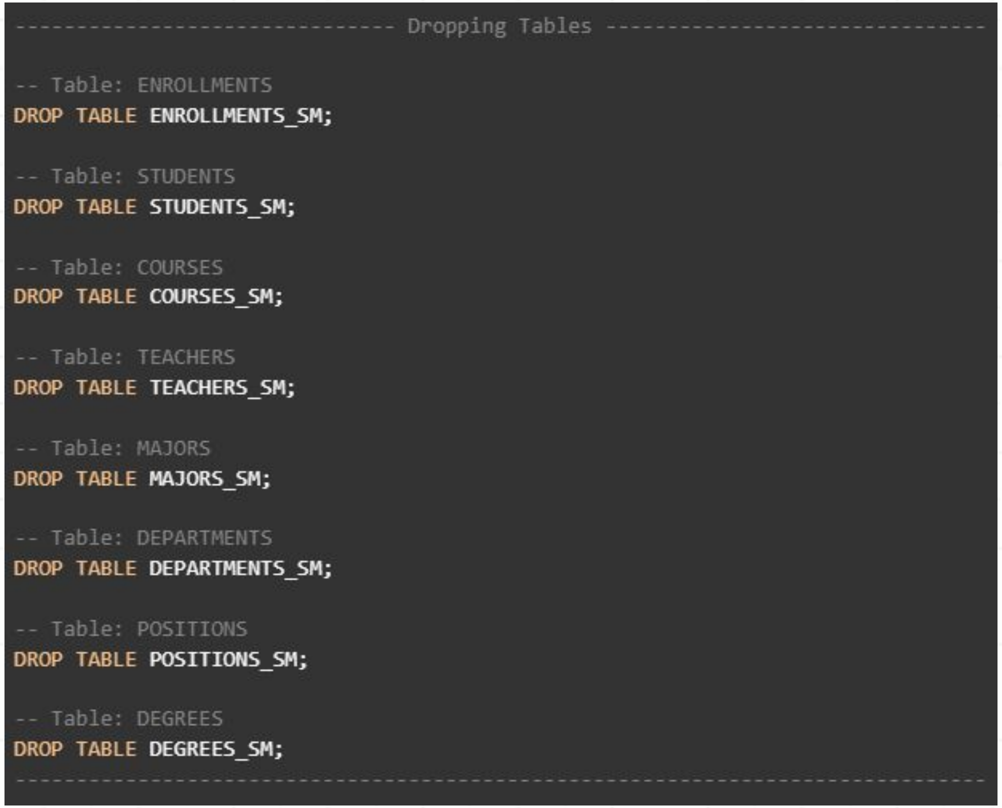
```
------------------------------ Clearing Records -------------------------------

-- Table: ENROLLMENTS
TRUNCATE TABLE ENROLLMENTS_SM;

-- Table: STUDENTS
TRUNCATE TABLE STUDENTS_SM;

-- Table: COURSES
TRUNCATE TABLE COURSES_SM;

-- Table: TEACHERS
TRUNCATE TABLE TEACHERS_SM;

-- Table: MAJORS
TRUNCATE TABLE MAJORS_SM;

-- Table: DEPARTMENTS
TRUNCATE TABLE DEPARTMENTS_SM;

-- Table: POSITIONS
TRUNCATE TABLE POSITIONS_SM;

-- Table: DEGREES
TRUNCATE TABLE DEGREES_SM;


COMMIT;
-------------------------------------------------------------------------------
```

drop_constraints.sql



```sql
---------------------------- Dropping Constraints ----------------------------

-- Table: ENROLLMENTS
ALTER TABLE ENROLLMENTS_SM
DROP CONSTRAINT enrollment_student_reference;

ALTER TABLE ENROLLMENTS_SM
DROP CONSTRAINT enrollment_course_reference;


-- Table: STUDENTS
ALTER TABLE STUDENTS_SM
DROP CONSTRAINT student_major_reference;


-- Table: COURSES
ALTER TABLE COURSES_SM
DROP CONSTRAINT course_instructor_reference;


-- Table: TEACHERS
ALTER TABLE TEACHERS_SM
DROP CONSTRAINT teacher_position_reference;

ALTER TABLE TEACHERS_SM
DROP CONSTRAINT teacher_department_reference;


-- Table: MAJORS
ALTER TABLE MAJORS_SM
DROP CONSTRAINT major_degree_reference;

ALTER TABLE MAJORS_SM
DROP CONSTRAINT major_department_reference;
--------------------------------------------------------------------------------
```

drop_tables.sql



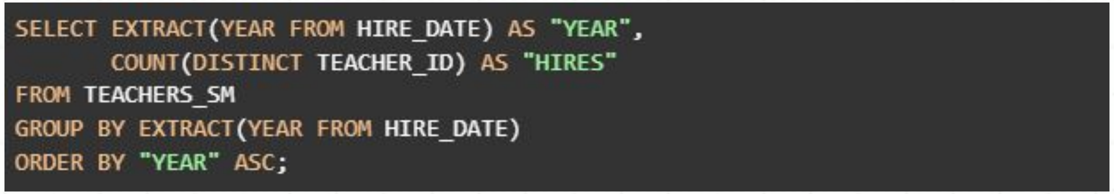
```
------------------------------- Dropping Tables -------------------------------

-- Table: ENROLLMENTS
DROP TABLE ENROLLMENTS_SM;

-- Table: STUDENTS
DROP TABLE STUDENTS_SM;

-- Table: COURSES
DROP TABLE COURSES_SM;

-- Table: TEACHERS
DROP TABLE TEACHERS_SM;

-- Table: MAJORS
DROP TABLE MAJORS_SM;

-- Table: DEPARTMENTS
DROP TABLE DEPARTMENTS_SM;

-- Table: POSITIONS
DROP TABLE POSITIONS_SM;

-- Table: DEGREES
DROP TABLE DEGREES_SM;
-------------------------------------------------------------------------------
```

04

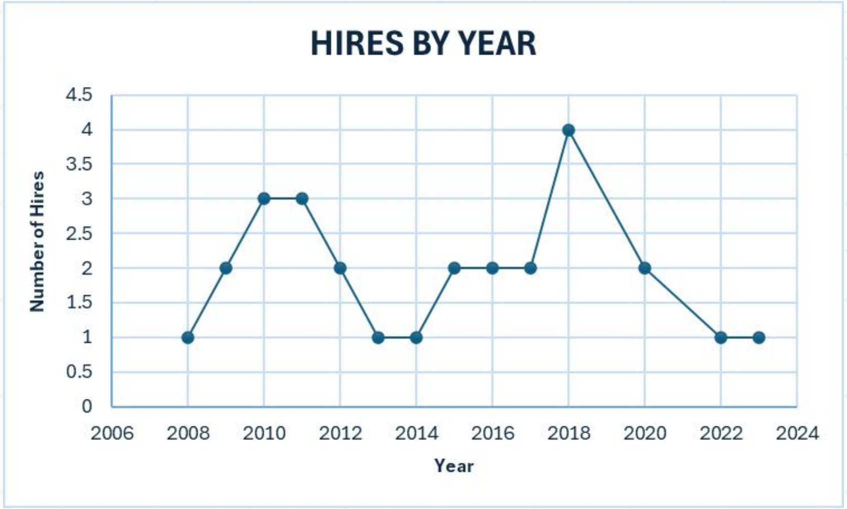
Data Querying & Analysis

Executing SQL queries (DQL) to extract meaningful data, analyze patterns, and derive insights from the database.

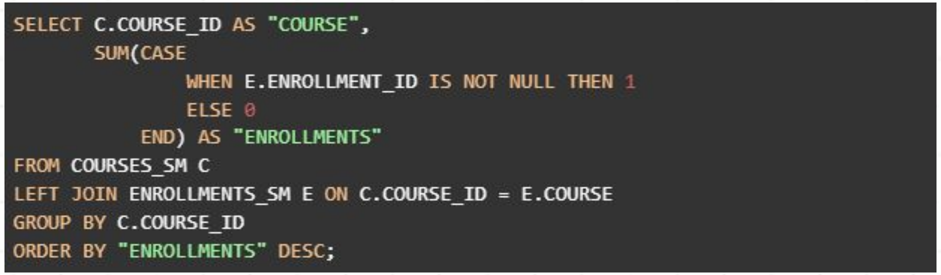
Distribution of hires by year



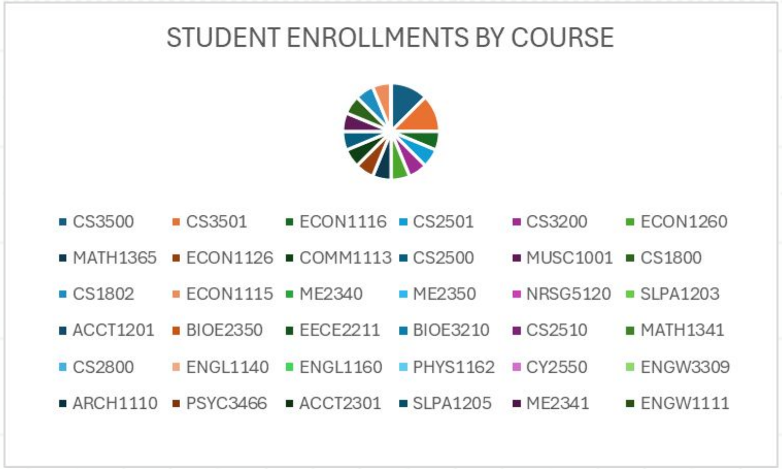
```
SELECT EXTRACT(YEAR FROM HIRE_DATE) AS "YEAR",
       COUNT(DISTINCT TEACHER_ID) AS "HIRES"
FROM TEACHERS_SM
GROUP BY EXTRACT(YEAR FROM HIRE_DATE)
ORDER BY "YEAR" ASC;
```



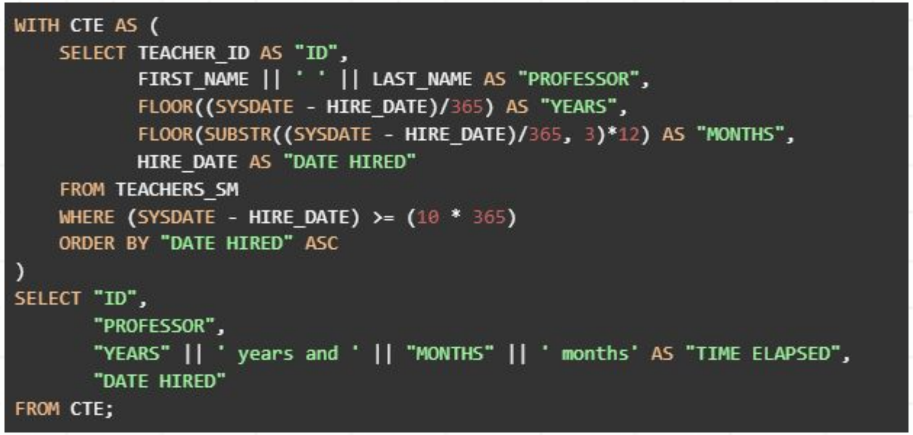
Distribution of student enrollments by course



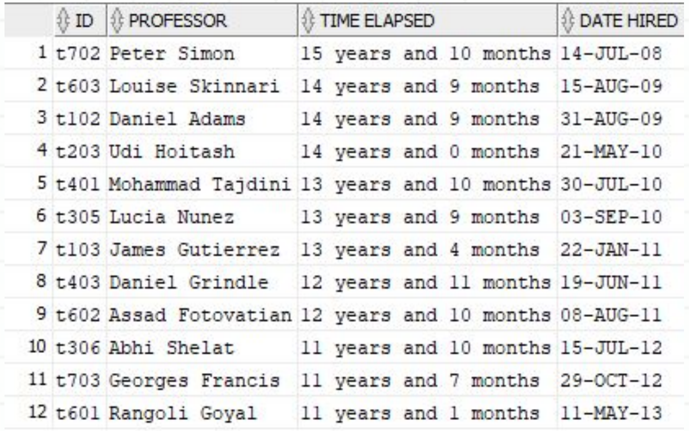
```sql
SELECT C.COURSE_ID AS "COURSE",
       SUM(CASE
               WHEN E.ENROLLMENT_ID IS NOT NULL THEN 1
               ELSE 0
           END) AS "ENROLLMENTS"
FROM COURSES_SM C
LEFT JOIN ENROLLMENTS_SM E ON C.COURSE_ID = E.COURSE
GROUP BY C.COURSE_ID
ORDER BY "ENROLLMENTS" DESC;
```



Teachers that have been hired for >= 10 years

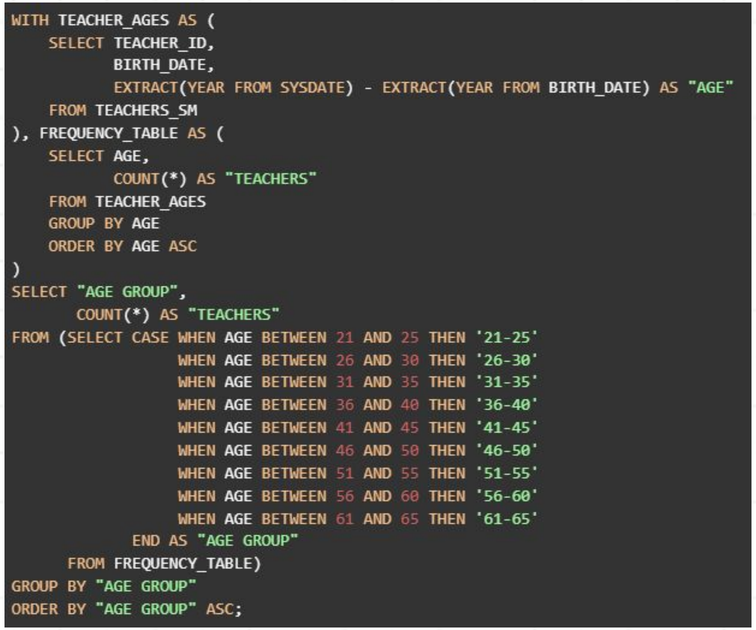


```sql
WITH CTE AS (
    SELECT TEACHER_ID AS "ID",
           FIRST_NAME || ' ' || LAST_NAME AS "PROFESSOR",
           FLOOR((SYSDATE - HIRE_DATE)/365) AS "YEARS",
           FLOOR(SUBSTR((SYSDATE - HIRE_DATE)/365, 3)*12) AS "MONTHS",
           HIRE_DATE AS "DATE HIRED"
    FROM TEACHERS_SM
    WHERE (SYSDATE - HIRE_DATE) >= (10 * 365)
    ORDER BY "DATE HIRED" ASC
)
SELECT "ID",
       "PROFESSOR",
       "YEARS" || ' years and ' || "MONTHS" || ' months' AS "TIME ELAPSED",
       "DATE HIRED"
FROM CTE;
```

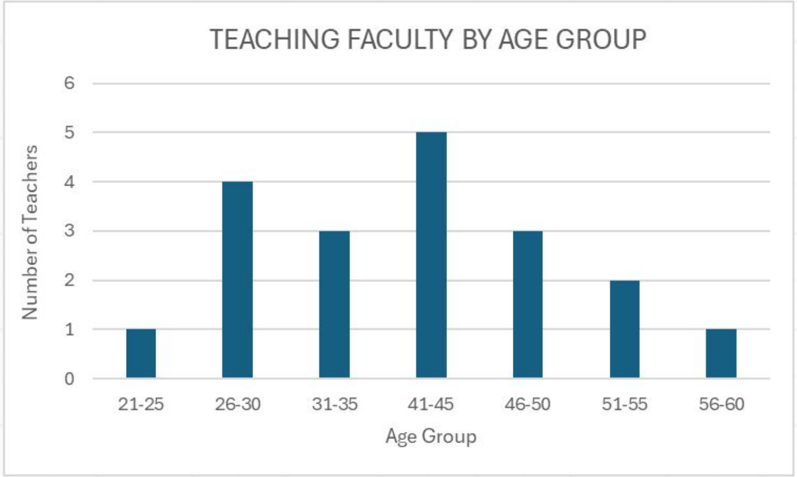


	ID	PROFESSOR	TIME ELAPSED	DATE HIRED
1	t702	Peter Simon	15 years and 10 months	14-JUL-08
2	t603	Louise Skinnari	14 years and 9 months	15-AUG-09
3	t102	Daniel Adams	14 years and 9 months	31-AUG-09
4	t203	Udi Hoitash	14 years and 0 months	21-MAY-10
5	t401	Mohammad Tajdini	13 years and 10 months	30-JUL-10
6	t305	Lucia Nunez	13 years and 9 months	03-SEP-10
7	t103	James Gutierrez	13 years and 4 months	22-JAN-11
8	t403	Daniel Grindle	12 years and 11 months	19-JUN-11
9	t602	Assad Fotovatian	12 years and 10 months	08-AUG-11
10	t306	Abhi Shelat	11 years and 10 months	15-JUL-12
11	t703	Georges Francis	11 years and 7 months	29-OCT-12
12	t601	Rangoli Goyal	11 years and 1 months	11-MAY-13

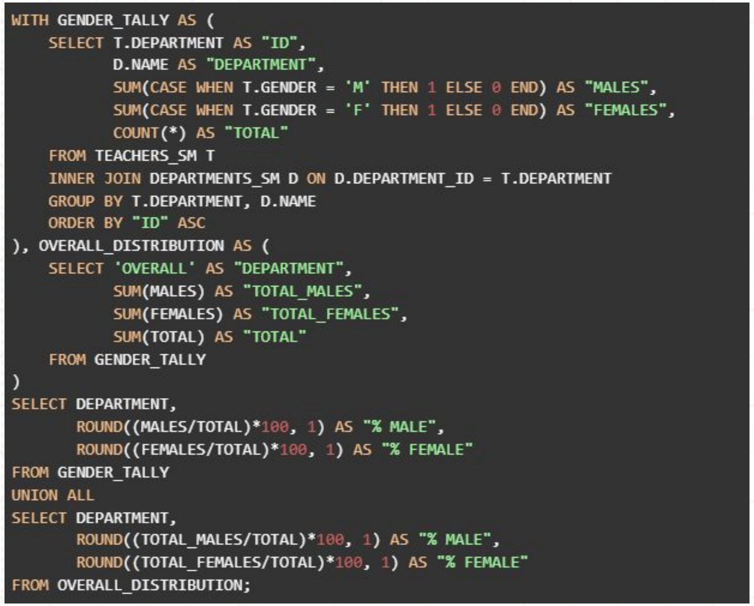
Distribution of teaching faculty by age



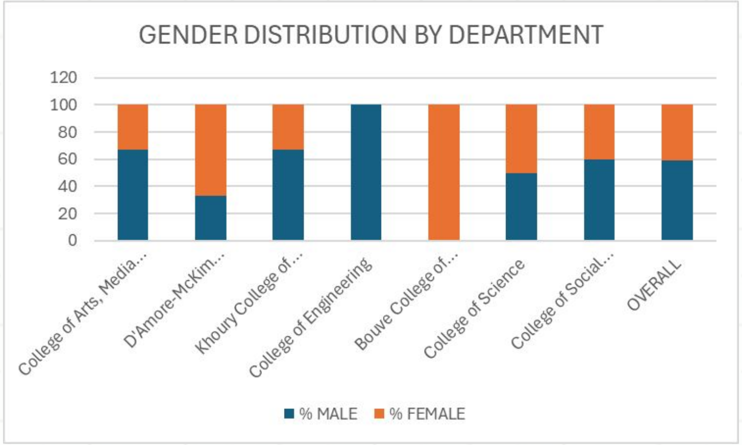
```
WITH TEACHER_AGES AS (
    SELECT TEACHER_ID,
           BIRTH_DATE,
           EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM BIRTH_DATE) AS "AGE"
    FROM TEACHERS_SM
), FREQUENCY_TABLE AS (
    SELECT AGE,
           COUNT(*) AS "TEACHERS"
    FROM TEACHER_AGES
    GROUP BY AGE
    ORDER BY AGE ASC
)
SELECT "AGE GROUP",
       COUNT(*) AS "TEACHERS"
FROM (SELECT CASE WHEN AGE BETWEEN 21 AND 25 THEN '21-25'
                  WHEN AGE BETWEEN 26 AND 30 THEN '26-30'
                  WHEN AGE BETWEEN 31 AND 35 THEN '31-35'
                  WHEN AGE BETWEEN 36 AND 40 THEN '36-40'
                  WHEN AGE BETWEEN 41 AND 45 THEN '41-45'
                  WHEN AGE BETWEEN 46 AND 50 THEN '46-50'
                  WHEN AGE BETWEEN 51 AND 55 THEN '51-55'
                  WHEN AGE BETWEEN 56 AND 60 THEN '56-60'
                  WHEN AGE BETWEEN 61 AND 65 THEN '61-65'
             END AS "AGE GROUP"
      FROM FREQUENCY_TABLE)
GROUP BY "AGE GROUP"
ORDER BY "AGE GROUP" ASC;
```



Distribution of teaching faculty by gender



```sql
WITH GENDER_TALLY AS (
    SELECT T.DEPARTMENT AS "ID",
           D.NAME AS "DEPARTMENT",
           SUM(CASE WHEN T.GENDER = 'M' THEN 1 ELSE 0 END) AS "MALES",
           SUM(CASE WHEN T.GENDER = 'F' THEN 1 ELSE 0 END) AS "FEMALES",
           COUNT(*) AS "TOTAL"
    FROM TEACHERS_SM T
    INNER JOIN DEPARTMENTS_SM D ON D.DEPARTMENT_ID = T.DEPARTMENT
    GROUP BY T.DEPARTMENT, D.NAME
    ORDER BY "ID" ASC
), OVERALL_DISTRIBUTION AS (
    SELECT 'OVERALL' AS "DEPARTMENT",
           SUM(MALES) AS "TOTAL_MALES",
           SUM(FEMALES) AS "TOTAL_FEMALES",
           SUM(TOTAL) AS "TOTAL"
    FROM GENDER_TALLY
)
SELECT DEPARTMENT,
       ROUND((MALES/TOTAL)*100, 1) AS "% MALE",
       ROUND((FEMALES/TOTAL)*100, 1) AS "% FEMALE"
FROM GENDER_TALLY
UNION ALL
SELECT DEPARTMENT,
       ROUND((TOTAL_MALES/TOTAL)*100, 1) AS "% MALE",
       ROUND((TOTAL_FEMALES/TOTAL)*100, 1) AS "% FEMALE"
FROM OVERALL_DISTRIBUTION;
```



Results of DQL Exploratory Analysis

- Lists:
    - Horizontal bar charts
        - Style: *"Top 10 […]s to …"*
- Distributions:
    - Histogram / Bar Charts
        - Density curves to fine-tune distribution shape#
        - Stacked bar charts for multiple groups
    - Pie Charts
        - Annotated sections
        - Comprehensive key and color-coordination
- Quantitative data:
    - Box plots
    - Scatter plots

Thanks For Listening