

## SQL Skill Based Assessment (SBA)

**Objectives:** This exercise is to test your skills with Databases and Structured Query Language (SQL). It will utilize your knowledge of SQL Data Definition Language (DDL), Data Modification Language (DML), and Data Query Language (DQL); most emphasis is placed on the DQL portion.

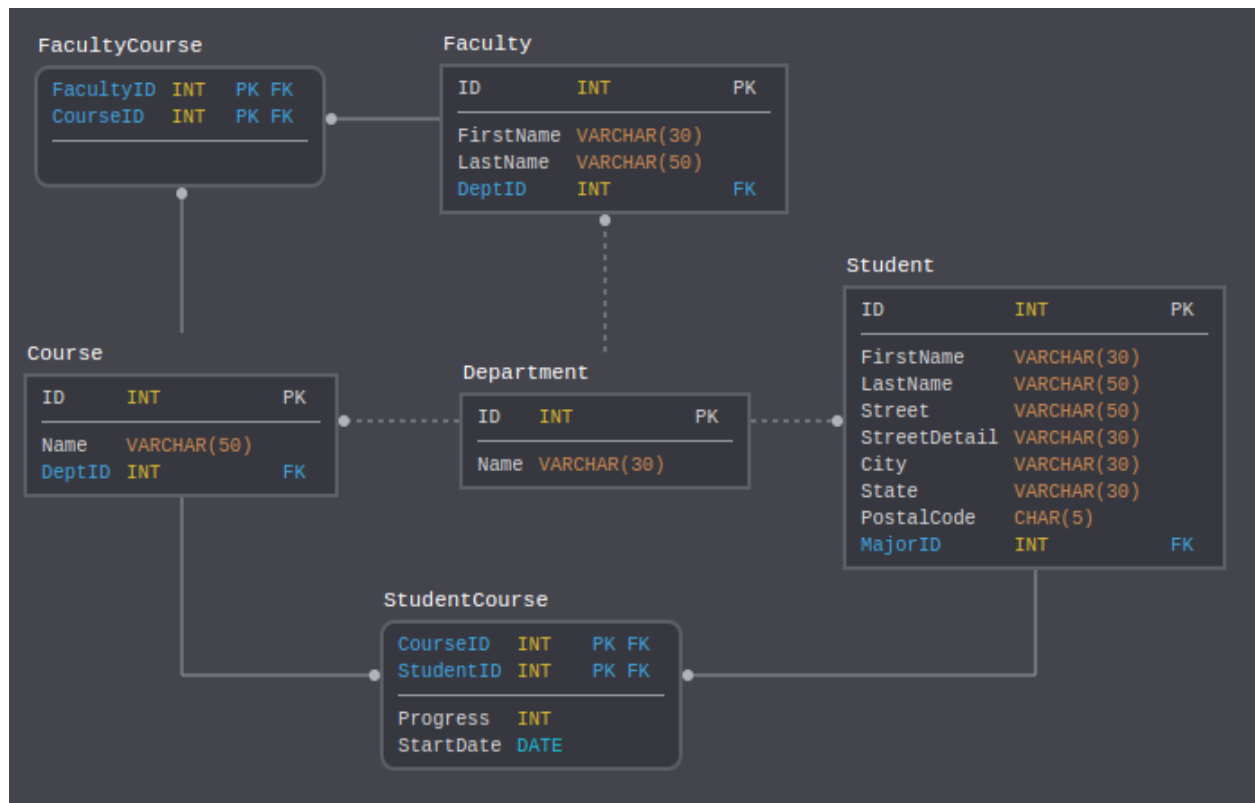
**Prerequisites:** You will be provided with a script **sba\_inserts.sql** which will populate the tables.

**Deliverables:** When you complete this assessment, you will provide two sql script files named **sba\_schema.sql** and **sba\_queries.sql**. These scripts will contain (in order) all of the select statements that you will write for steps 1 and 4 of this exercise.

### Scoring

1. All scripts must run.
2. **The scripts must be formatted for readability.** Use indentation and line-returns to make each statement readable. Poor readability may discount as much as 10% of the total possible score.
3. For DDL statements, the created tables must match the expected schemas. All field names and types must be correct, and all keys (primary and foreign) must be defined correctly.
4. For DQL statements, the expected output must be matched exactly. Extra, missing, or incorrectly-named columns are discounted. The number of results for each query must be as expected, and when ordering is specified, the ordering of rows must be correct.
5. Use of comments embedded into your scripts is encouraged if those comments make the script easier to understand.

## College Management System Schema



### Step 1: Apply the CREATE scripts

The provided CREATE script (osba.schema.sql) data must be applied to your database.

### Step 2: Apply the INSERT statements.

The provided INSERT script (sba\_insert.sql) data must be applied to your database.

You will not be able to proceed with this assignment until all INSERT statements succeed.

### Step 3: Apply the Alter statements

Perform the following changes to the StudentCourse table:

- Add a column named EndDate of type Date and a column named Credits of type INT.
- Add NOT NULL constraint to the column EndDate.
- Modify the name of the field EndDate to FinishDate.
- Write a query to remove the columns FinishDate and Credits from the StudentCourse table.

**Step 4: Write queries that return the following result sets:**

- A. The Curriculum Planning Committee is attempting to fill in gaps in the current course offerings. You need to provide them with a query which lists each department and the number of courses offered by that department. The two columns should have headers “Department Name” and “# Courses”, and the output should be sorted by “# Courses” in each department (ascending).

Department Name	# Courses
1 Information Technology	4
2 Social Sciences	5
3 Arts & Design	7
4 Business	8
5 Engineering	9
6 Sciences	13
7 Medicine	14
8 Culture & History	16

- B. The recruiting department needs to know which courses are most popular with the students. Please provide them with a query which lists the name of each course and the number of students in that course. The two columns should have headers “Course Name” and “# Students”, and the output should be sorted first by # Students descending and then by course name ascending.

Course Name	# Students
Fashion	47
Robotics	45
Computer Science	43
Librarianship & Information Management	40
Drama, Dance & Cinematics	39
Education	39
Social Work	38
Creative Writing	37
Art & Design	36

- C. Quite a few students have been complaining that the professors are absent from some of their courses.
1. Write a query to list the names of all courses where the # of faculty assigned to those courses is zero. The output should be in alphabetical order by course name.

NAME
Archaeology
Celtic Studies
Complementary Medicine
Counselling
East & South Asian Studies
English
Forensic Science
General Engineering
German

2. Using the above, write a query to list the course names and the # of students in those courses for all courses where there are no assigned faculty. The output should be ordered first by # of students descending and then by course name ascending.

NAME	#_student
Psychology	33
General Engineering	27
Archaeology	26
Forensic Science	26
Counselling	22
East & South Asian Studies	21
English	20
Celtic Studies	19
Complementary Medicine	19

- D. The enrollment team is gathering analytics about student enrollment throughout the years. Write a query that lists the total # of students that were enrolled in classes during each school year. The first column should have the header “Students”. Provide a second “Year” column showing the enrollment year. Display the records sorted in ascending order based on startDate.

STUDENTS	YEAR
145	2014
52	2014
114	2015
300	2015
363	2016
145	2016
555	2017
317	2017

- E. The enrollment team is gathering analytics about student enrollment and they now want to know about August admissions specifically. Write a query that lists the Start Date and # of Students who enrolled in classes in August of each year. Output should be ordered by start date ascending.

Start Date	# of Students
25-AUG-14	52
25-AUG-15	114
25-AUG-16	145
25-AUG-17	317

- F. Students are required to take 4 courses, and at least two of these courses must be from the department of their major. Write a query to list students' First Name, Last Name, and number of courses they are taking in their major department. The output should be sorted first in increasing order of the number of courses, then by student last name. Use aliases "First Name", "Last Name" and "Number of Courses".

First Name	Last Name	Number of Courses
Caryn	Abdallah	2
Lashanda	Abdul	2
Wilburn	Accardo	2
Jenna	Ackles	2
Zella	Acklin	2
Elaina	Alers	2
Maranda	Armand	2
Lura	Atilano	2
Johanne	Bacote	2

- G. Students making average progress in their courses of less than 50% need to be offered tutoring assistance. Write a query to list First Name, Last Name and Average Progress of all students achieving average progress of less than 50%. The average progress *as displayed* should be rounded to one decimal place. Sort the output by average progress descending. Use aliases "First Name", "Last Name" and "Average Progress".

First Name	Last Name	Average Progress
Alia	Stenson	49.8
Quiana	Thompson	49.3
Teri	Hartzler	49.3
Gwendolyn	Wentworth	49.3
Debera	Butner	49
France	Daoust	48.8
Leeanna	Theiss	48.7
Russel	Neloms	48.5
Wilhelmina	McAuliffe	48.3

- H. Faculty are awarded bonuses based on the progress made by students in their courses.
- Write a query to list each course name and the average progress of students in that course. The output should be sorted descending by average progress. Use aliases "Course Name" and "Average Progress".

Course Name	Average Progress
Linguistics	69.76470588235294...
Education	69.56410256410256...
Land & Property Management	69.16
Counselling	67.81818181818181...
Veterinary Medicine	67.15
Politics	67.05263157894736...
Medicine	66.83333333333333...
Russian & East European Languages	66.32
Materials Technology	66.21428571428571

- Write a query that selects the maximum value of the average progress reported by the previous query.

MAX(AVG_PROG)
69.76470588235294117647058823529411764706

- Write a query that outputs the faculty First Name, Last Name, and average of the progress made over all of their courses. Use aliases "First Name", "Last Name" and "Avg. Progress".

First Name	Last Name	Average Progress
Jadwiga	Braud	58.275
Will	Eshbaugh	58.4666666666...
Katheleen	Matinez	58.5172413793...
Marguerita	Bogert	59.0252100840...
Malena	Antonucci	60.0666666666...
Roosevelt	Hruska	60.3831775700...
Gaylene	Roten	60.4210526315...
Karol	Dicicco	60.5762711864...
Erha	Archer	60.5882352941

- Write a query just like #3, but where only those faculties where average progress in their courses is 90% or more of the maximum observed in #2.

Display the records sorted in descending order based on "Average Progress".

First Name	Last Name	Average Progress
Corrin	Tomczak	65.98809523809...
Rina	Hornsby	65.67142857142...
Fanny	Ewell	64.68831168831...
Cordie	Farrow	64.62711864406...
Ma	Roig	64.62025316455...
Dolly	Farago	64.55696202531...
Jere	Balch	64.45714285714...
Shirley	Salomone	64.29268292682...
Doris	Budde	64.13684210526

- Students are awarded two grades based on the minimum and maximum progress they are making in the courses. The grading scale is as follows:

Progress < 40:	F
Progress < 50:	D
Progress < 60:	C
Progress < 70:	B
Progress >= 70:	A

Write a query which displays each student's "First Name", "Last Name", minimum progress with "Min Grade" as alias, and maximum progress with alias "Max Grade". Display the records sorted in ascending order based on "Average Progress".

FIRST NAME	LAST NAME	Min Grade	Max Grade
Rosena	Kulinski	F	D
Ronny	Dublin	F	D
Marlyn	Segers	F	C
Maxine	Restivo	F	D
Edyth	Trostle	F	C
Antony	Chess	F	C
Angeles	Lamont	F	C
Nathalie	Hudock	F	C
Britt	Brumit	F	C

- J. Write a query that returns student's full name with "Student Name" as alias whose progress is greater than the average progress for their course and order by "Student Name"

Student Name
Ada Caul
Adah Clark
Adan Dabbs
Adelaide Coonrod
Adelia Rew
Adolph Pavlik
Alan Demartino
Alanna Kimbrell
Albertina Hirsh