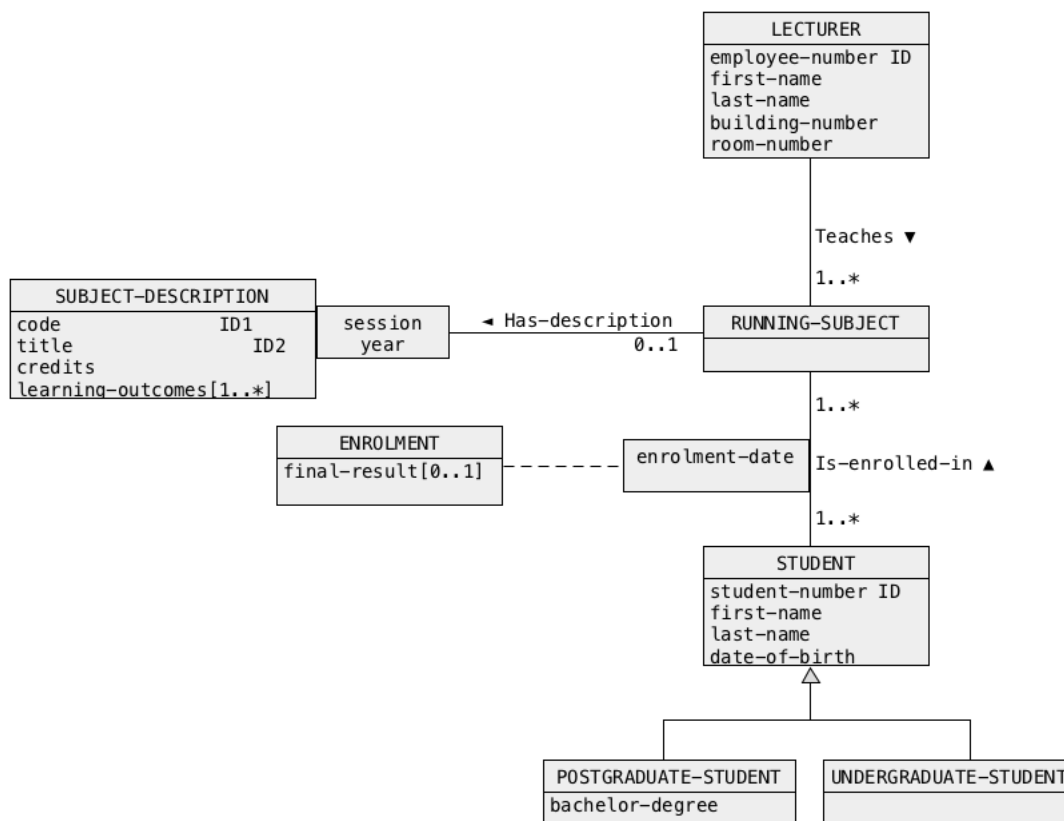


CSCI835 Database Systems (Bridging Subject)
Practice
22 July 2019

Task 1

CREATE TABLE statement

Consider a conceptual schema given below.



Perform logical database design and transform a conceptual schema given above into a collection of relational schemas. No report is expected from logical design. Use a **superset** method for implementation of generalization.

Next, use the relational schemas obtained in the previous step to create SQL script solution1.sql with CREATE TABLE statements that implement the relational schemas.

Note, that you MUST use only CREATE TABLE statements and no other statement of SQL.

When a script file `solution1.sql` is ready connect to Oracle either through command line interface SQLPlus or graphical user interface SQL Developer and process your script file.

To list the names of relational tables created you can use a script `list.sql`.

If processing of the file returns the errors then you must eliminate the errors! Processing of your script must return NO ERRORS !

It is recommended to create a script `drop.sql` that drops all relational tables created by processing of a script `solution1.sql` and it is recommend to execute `drop.sql` after each processing of `solution1.sql`

Task 2

ALTER TABLE statement

Download the files `dbcreate.sql`, `dbdrop.sql`, and `dbschema.pdf`.

Connect to Oracle database server either through command line interface SQLPlus or graphical user interface SQL Developer.

To create and to load the relational tables of a sample database, process SQL script `dbcreate.sql`. A script `dbdrop.sql` can be used to drop the relational tables. Do not drop the relational tables now.

Use a pdf viewer to open a file `dbschema.pdf` with a conceptual schema of the sample database. The green blobs represent the relational tables that implement the classes of objects and associations.

Your task is to create and to process SQL script `solution2.sql` with only `ALTER TABLE` statements that implement the following modifications of the structures of relational tables included in the sample database. Of course, you do not need to modify the contents of the relational tables. Note, that some modifications may require more than one `ALTER TABLE` statement.

- (1) We would like to be able to add information about the total number of applications submitted by each applicant. Assume that, an applicant cannot submit more than 3 applications at a time.
- (2) We would like to decrease the maximum length of specification in `POSITION` table up to and including 1500 characters;
- (3) We would like to increase the maximum bonus of a position up to 99999999.99 in any currency.
- (4) We would like to add an association `APPLICANT` Is-an-expert-in `SKILL` to the database. The association Is-expert-in is "one-to-many". It means that an applicant is an expert in at most one skill and a skill may have zero or more experts.
- (5) In the future there is no need to store information about the numbers of faxes of the applicants and the fax numbers already entered into the database can be lost.

When SQL script file `solution2.sql` is ready process your script file.

It is recommended to use a script `dbdrop.sql` to drop all relational tables modified during the processing of a script `solution2.sql` and then to re-create the original

database with a script `dbcreate.sql`. In such a way your script always operates on the original structures of the sample database.

Task 3

Simple data manipulation statements

Download the files `dbcreate.sql`, `dbdrop.sql`, and `dbschema.pdf`.

Connect to Oracle database server either through command line interface SQLPlus or graphical user interface SQL Developer.

To create and to load the relational tables of a sample database, process SQL script `dbcreate.sql`. A script `dbdrop.sql` can be used to drop the relational tables. Do not drop the relational tables now.

Use a pdf viewer to open a file `dbschema.pdf` with a conceptual schema of the sample database. The green blobs represent the relational tables that implement the classes of objects and associations.

Your task is to use `INSERT` and/or `DELETE` and/or `UPDATE` statements of SQL to implement a script file `solution3.sql` that performs the data manipulations listed below. You must drop all tables of the sample database, re-create all tables of the database and load data into the database with `dbcreate.sql` script before implementation of this task.

Your SQL statements must operate on the sample database loaded with data.

An important condition is that you must only use `INSERT` and/or `DELETE` and/or `UPDATE` statements of SQL. No other statements of SQL are allowed. It means that you are not allowed to change any consistency constraints imposed on the contents of the data base like, for example suspension of foreign key constraints, etc.

Note, that implementation of the actions listed below may require more than one SQL statement.

- (1) Insert into the database information about a new skill possessed by an applicant number 7. The new skill has a name `climbing` and it needs a skill `thinking`. The applicant possesses the new skill at a level 3.
- (2) A position number 1 is no longer required. Remove all information about the position from the database and all information related to the position.
- (3) Due to an earthquake all applicants living in `Melbourne` have been evacuated to `Sydney`. Their phone numbers have not changed, however, their fax numbers and email addresses are no longer valid and we do not know the new values. Update the database.

- (4) A name of skill `thinking` must be changed to `Deep thinking`. Modify the database.
-

Task 4

SELECT statement: statements with WHERE, GROUP BY and HAVING clauses

Download the files `dbcreate.sql`, `dbdrop.sql`, and `dbschema.pdf`.

Connect to Oracle database server either through command line interface SQLPlus or graphical user interface SQL Developer.

To create and to load the relational tables of a sample database, process SQL script `dbcreate.sql`. A script `dbdrop.sql` can be used to drop the relational tables. Do not drop the relational tables now.

Use a pdf viewer to open a file `dbschema.pdf` with a conceptual schema of the sample database. The green blobs represent the relational tables that implement the classes of objects and associations.

Your task is to implement the following queries as `SELECT` statements of SQL and save the statements in SQL script file `solution4.sql`.

- (1) Find the name of employers located in Perth, Western Australia.
- (2) Find the numbers and titles of positions with salary in a range between 200000 and 800000 inclusive.
- (3) Find the employers that have either email address or web site address and not both email address and web site address. List the names of employers, email addresses and web site addresses.
- (4) Find the first and last names of all applicants born in 1960. It is your task to find an appropriate standard function that can be used to extract a year from a date.
- (5) List the numbers and titles of all positions. The results should be displayed in the ascending order of salaries and for all positions with the same salary in the descending order of titles.
- (6) Find the total number of all positions offered, average salary of all position, the largest and the smallest salary. This query must be implemented as a single `SELECT` statement.
- (7) Find the names of employers together with total number of positions offered by each one of them. You may ignore the employers who offer no positions at the moment.
- (8) For each one of the skills C programming, C++ programming and SQL programming find the total number of applicants that possess such skill at a level greater than 5. List a name of skill and the total number of applicants that possess such skill at a level greater than 5.

(9) For each applicant list his/her number and the total number of skills possessed. List only the applicants that possess more than 3 skills.

(10) Find the names of all skills that are possessed by at least one applicant or are needed for at least one position. Do not list the identical names many times.

Task 5

SELECT statement: joins, antijoins, GROUP BY and HAVING clauses

Download the files `dbcreate.sql`, `dbdrop.sql`, and `dbschema.pdf`.

Connect to Oracle database server either through command line interface SQLPlus or graphical user interface SQL Developer.

To create and to load the relational tables of a sample database, process SQL script `dbcreate.sql`. A script `dbdrop.sql` can be used to drop the relational tables. Do not drop the relational tables now.

Use a pdf viewer to open a file `dbschema.pdf` with a conceptual schema of the sample database. The green blobs represent the relational tables that implement the classes of objects and associations.

Your task is to implement the following queries as `SELECT` statements of SQL, save and save the statements in SQL script file `solution5.sql` and process the script file.

The queries listed below must be implemented as `SELECT` statements with `JOIN` operation.

- (1) Find the titles of positions offered by the employers from Western Australia.
- (2) Find the first and last names of applicants who applied for a position offered by University of Queensland.

The queries listed below must be implemented as nested `SELECT` statements with `IN/NOT IN` set membership operation.

- (3) Find the titles of positions offered by the employers from Western Australia.
- (4) Find the first and last names of all applicants who submitted at least one application.
- (5) Find the first and last names of all applicants who submitted no applications.
- (6) Find the first and last names of applicants who submitted exactly one application.

The queries listed below must be implemented as nested queries with `EXISTS/NOT EXISTS` clauses.

- (7) Find the titles of positions offered by the employers from Western Australia.
- (8) Find the first and last names of all applicants who submitted at least one application.

(9) Find the first and last names of all applicants who submitted no applications.

(10) Find the first and last names of applicants who submitted exactly one application.

Task 6

Advanced data manipulations

Download the files `dbcreate.sql`, `dbdrop.sql`, and `dbschema.pdf`.

Connect to Oracle database server either through command line interface SQLPlus or graphical user interface SQL Developer.

To create and to load the relational tables of a sample database, process SQL script `dbcreate.sql`. A script `dbdrop.sql` can be used to drop the relational tables. Do not drop the relational tables now.

Use a pdf viewer to open a file `dbschema.pdf` with a conceptual schema of the sample database. The green blobs represent the relational tables that implement the classes of objects and associations.

Your task is to implement the following advanced manipulations on data in SQL, save your implementations in SQL script file `solution6.sql` and process the script.

- (1) Create an empty relational table to store information about the names of employers and the total number of positions offered by each employer. A name of relational table and the names of columns are up to you. Enforce the appropriate primary key and referential integrity constraints (if any).

Next, load into the table information about the names of employers together with the total number of positions offered by each employer. If an employer offers no position then its name must be recorded with a number zero (0).

- (2) Create a relational table that contains information about the names of all skills and the largest skill level possessed by an applicant and a number of applicant who possesses a skill at the highest level. Ignore the skills not possessed by any applicant.

All data must be loaded into the table by the same SQL statement that creates the table.

Enforce the appropriate primary key and referential integrity constraints (if any) after data is loaded.

- (3) Add a column to a relational table `POSITIONS` to store information about the total number of skills needed by each advertised position. A name of the column is up to you. Assume that no more than 9 skills are needed for each position.

Next, use a single `UPDATE` statement to set the values in the new column consistent with the present contents of the sample database.

- (4) Use a single `DELETE` statement to remove from a relational table `APPLIES` all applications submitted by the applicants who do not have any skills. `DELETE` statement must be correct for any contents of the sample database.

Task 7

Relational views

Download the files `dbcreate.sql`, `dbdrop.sql`, and `dbschema.pdf`.

Connect to Oracle database server either through command line interface SQLPlus or graphical user interface SQL Developer.

To create and to load the relational tables of a sample database, process SQL script `dbcreate.sql`. A script `dbdrop.sql` can be used to drop the relational tables. Do not drop the relational tables now.

Use a pdf viewer to open a file `dbschema.pdf` with a conceptual schema of the sample database. The green blobs represent the relational tables that implement the classes of objects and associations.

Create SQL script `solution7.sql` that implements of the following query in SQL.

Find full names (first name, last name) and an average skill level of all applicants whose average skill level is below an average skill level of all skills possessed.

For example, if applicant X has three skills at the levels 5, 5, and 8 and applicant Y has one skill at level 2 and applicant Z has no skills then an average skill level of all skills possessed is $(5+5+8+2)/4 = 5$. An average skill level of applicant X is $(5+5+8)/3 = 6$. An average skill level of applicant Y is $2/1 = 2$. An average skill level of applicant Z is 0. Therefore, the full names of applicants Y and Z should be listed together with average skill level for each one of them.

Y	2
Z	0

The query must be implemented in the following way.

- (1) First, create a relational view that contains information about an average skill level of all skills possessed by at least one applicant. A name of relational view is up to you.
- (2) Next, create a relational view that for each applicant who possesses at least one skill contains information about an applicant number and an average level of all his/her skills. Remember, that the view must contain information about the applicants who possess at least one skill. A name of relational view is up to you.
- (3) Next, create a relational view that for each applicant who possesses no skills contains information about an applicant number and 0 as his/her an average level of all skills. A name of relational view is up to you.
- (4) Next, create a relational view that is union of the views created in step (2) and step (3). A name of relational view is up to you.

- (5) Next, create a relational view that contains information about applicant numbers whose average skill level is below an average skill level computed in step (1). The new relational view created in this step must use information included in the views created in the steps (1) and (4). A name of relational view is up to you.
- (6) Finally, implement `SELECT` statement that uses a relational view created in a step (5) and a relational table `APPLICANT` to *find full names (first name, last name) and an average skill level of all applicants whose average skill level is below an average skill level of all skills possessed by at least one applicant.*

Include `CREATE VIEW` statements implementing the steps from (1) to (5) and the final `SELECT` statement into SQL script `solution7.sql` When ready process the script.

End of specification