Assignment for Module 3

The assignment in Module 3 provides experience with the CREATE TABLE statement for the Intercollegiate Athletic (ICA) Database. Note that this assignment extends the practice problems for the ICA database. You should use Oracle Cloud or PostgreSQL for this assignment. For each DBMS, you need to use a client to connect to the database server such as the SQL Developer for Oracle Cloud or pgAdmin for PostgreSQL.

1. Basic CREATE TABLE Statement Requirements

You should use the table descriptions in the Intercollegiate Database background document. You must use the same table and column names as specified in the background document. Here is advice about data type selections.

- You should use standard SQL data types specified in the notes. For columns with variable length text, you should use VARCHAR, the SQL standard data type.
- For primary key fields (*CustNo, LocNo, EventNo, PlanNo, EmpNo, ResNo,* and *FacNo*), use the VARCHAR data type with length 8. For consistency, corresponding foreign keys (such as *EventRequest.CustNo*) should also be the same data type and length.
- For Oracle, you should use the DATE data type for columns involving dates or times. The *EventPlanLine.TimeStart* and *EventPlanLine.TimeEnd* columns will store both date and time details so you should use the DATE data type. For PostgreSQL, you should use the DATE data type for columns with just date details (date columns in the *EventRequest* and *EventPlan* tables) and TIMESTAMP for columns with date and time details (time columns in the *EventPlanLine* table).

- Use CHAR(1) for the *Customer.Internal* column as Oracle does not provide a BOOLEAN data type. PostgreSQL has the BIT(1) data type, but I suggest that you use CHAR(1) instead.

2. Constraints

After writing the basic CREATE TABLE statements, you should modify the statements with constraints. The CONSTRAINT clauses can be either inline in a column definition or separate after column definitions except where noted. You should specify a meaningful name for each CONSTRAINT clause.

- For each primary key, you should specify a PRIMARY KEY constraint clause. For single column primary keys (*CustNo*, *LocNo*, *EventNo*, *PlanNo*, *EmpNo*, *ResNo*, and *FacNo*), the constraint clause can be inline or external. For multiple column primary keys (combination of *PlanNo* and *LineNo*), the CONSTRAINT clause must be external.
- For each foreign key, you should specify a FOREIGN KEY constraint clause. The constraint clauses can be inline or external.
- The foreign key constraint for *PlanNo* in the *EventPlanLine* table should cascade deletions. Cascading deletes means that deletion of a parent row in the *EventPlan* table causes deletion of related rows in the *EventPlanLine* table. Hint: use the ON DELETE CASCADE clause.
- Define NOT NULL constraints for all columns except *EventPlan.EmpNo*, *EventRequest.DateAuth*, *EventRequest.BudNo*, and *EventPlan.Notes*. NOT NULL constraints for the PK of each table are optional as not null is implied with a primary key constraint.
- Define a named CHECK constraint to restrict the *EventRequest.Status* column to have a value of “Pending”, “Denied”, or “Approved”. You can use the IN operator in this constraint.

- Define named CHECK constraints to ensure that *ResourceTbl.Rate* and *EventRequest.EstAudience* are greater than 0.
- Define a named CHECK constraint involving *EventPlanLine.TimeStart* and *EventPlanLineTimeEnd*. The start time should be smaller (chronologically before) than the end time. This CHECK constraint must be external because it involves two columns.

3. Populating Tables

The course website contains a text file containing SQL INSERT statements to populate the tables depending on the database server. You need to create the tables before inserting rows in each table. You need to insert rows in parent tables before child tables that reference parent tables. The INSERT statements in the file are in a proper order for populating the tables.

4. Initial CREATE TABLE Statements

To facilitate your work, you can use the CREATE TABLE statements you wrote in the ICA practice problems of Module 3 for the *Customer*, *Facility*, and *Location* tables. Module 3 contains a document with solutions for the ICA practice problems. Thus, you only need to write CREATE TABLE statements for the remaining five tables (*ResourceTbl*, *Employee*, *EventRequest*, *EventPlan*, and *EventPlanLine*). You still need to execute the CREATE TABLE statements to create all the tables and the INSERT statements to populate all tables.

5. Submission

The submission requirements involve CREATE TABLE statements and evidence that you executed the statements and created the tables in Oracle or PostgreSQL. You should submit 1 document containing a CREATE TABLE statement and screen snapshot for each table. You should neatly format your CREATE TABLE statements. You should use the same table and column names as specified in the ICA database background document. For the screen snapshot, you need to capture a screen showing most columns and rows of the populated table. You can use a feature of the Oracle or PostgreSQL client to show the rows in a table. Alternatively, you can execute an SQL statement (for example, SELECT * FROM ResourceTbl) to show the columns and rows.