Semester Three Database Programming Sprint One

Authored by Peter Rawsthorne, peter.rawsthorne@keyin.com

Project Description:

Having the skills and knowledge to derive a database design from a business scenario is essential for the full stack software developer. In this project each team will select a business scenario from a collection provided by the instructor to build their database. The project includes a fully attributed Entity Relationship Diagram (ERD) brough to third normal form. The database design should then be used to create a database in PostgreSQL containing all the identified entities with all primary and foreign keys defined and implemented. Once deployed to PostgreSQL the database should be loaded with test data by using a data mocking tool like https://www.mockaroo.com/. SQL queries with JOINs should be written to exercise the database design against use cases identified from the written business description. Once all the project outcomes have been completed the database should be backed up and included with the database ER diagram and all the SQL queries for submission.

Learning Outcomes:

1. Proven ability to design a relational database from a written business scenario.
2. Proven ability to identify all the business entities from the business scenario.
3. Proven ability to fully attribute all entities with the fields (columns) to fulfill the requirements found in the business scenario.
4. Proven ability to create an Entity Relationship Diagram (ERD) modeling all the entities, attributes, and their relationships.
5. Proven ability to use a standard symbology to identify one-to-one, one-to-many, and many-to-many relationships.
6. Proven ability to model the entities and attributes to third normal form.
7. Proven ability to implement the database in PostgreSQL using diagramming tools and SQL CREATE statements.
8. Proven ability to load a database using mock data generation and written INSERT and UPDATE SQL statements.
9. Proven ability to backup and restore a database from one computer to another.
10. Proven ability to confirm their database design by exercising the data model with SQL queries.

Marking:

In this program core evaluation is marked with one of three possible marks: *Incomplete (Fail), Pass, and Pass Outstanding.* For Sprints and QAPs, where incomplete marks are more important for our own information as well as for the information of the student, we wanted to increase the resolution of our grading system. Therefore, each QAP and team-based Sprint have outcomes that are marked on a scale of 1-5. The details of this marking system are summarized in the table below. See project rubric found in the respective MS-Teams Assignments description.

Grade	Meaning
1	*Incomplete.* Student or Team shows severe lack of understanding of the material – solution is heavily incomplete, non-functional, or completely off base of what the assignment was asking for.
2	*Partially Complete.* Student or Team show some understanding of the material. Solution may be non-functional or partially functional, but the approach is correct, albeit with some major bugs or missing features.
3	*Mostly Complete.* Student or Team demonstrates understanding of the major ideas of the assignment. Solution is mostly working, albeit with a few small bugs or significant edge cases which were not considered. Shows a good understanding of the correct approach, and is either nearly a feature-complete solution, or is a feature-complete solution with some bugs.
4	*Complete (Equivalent to: Pass.)* Student or Team shows complete understanding of assigned work and implemented all necessary features. Any bugs that are present are insignificant (for example aesthetic bugs when testing the functionality of code) and do not impact the core functionality in a significant way. All necessary objectives for the assignment are completed, and the student or team has delivered something roughly equivalent to the canonical solution in terms of features and approach.
5	*Complete with Distinction (Equivalent to: Pass Outstanding)* The Student or Team demonstrates a clear mastery of the subject matter tested by the Sprint or QAP. The solution goes above and beyond in some way, makes improvements on the canonical solution, or otherwise demonstrates the student's mastery of the subject matter in some way. A solution in this category would consider all reasonable edge cases and implement more than the necessary functionality required by the assignment.

Reminder: See project rubric found in the respective MS-Teams Assignments description.

Project Scenarios:

As a team, choose one of the written business scenarios from the collection provided in the companion zip file. Choose two scenarios, your first and second choice. Email you chosen scenarios by number and title to your instructor as soon as you can. No two teams will be doing the same scenarios, so first come first served scenario selection to each team. Keep in mind these scenarios may have incomplete or incorrect identification of entities.

Project features as user stories:

Roles:

- **Database Administrators** (DBA) will keep the database up and running smoothly 24/7. They are responsible for design, implementation, and optimization of database systems. Their goal is to provide a seamless flow of information throughout the company, considering both backend infrastructure and frontend accessibility for end-users.
- **Business User** means a person who uses the software services and/or equipment in the course of any trade or business activity.

Stories:

- As a DBA I would like to have an ER diagram to assess the database design for data quality, referential integrity, infrastructure requirements, and database performance.
- As a DBA I would like to have a development instance of the database with sample data to assess the creation scripts, table relationships, and data usability.
- As a business user I would like to have pre-written SQL queries that provide the resulting data sets that fulfill my reporting and business analysis requirements. These queries should exercise all the possible JOINs within the database design.

Project Deliverables:

A single zip file that contains the following;

1. The original document describing the chosen business scenario.
2. A fully attributed ERD with all PK and FK relationships defined using a standard symbology.
3. CREATE statements saved to disk files for creating all entities, attributes, keys, and relationships.
4. A database backup including all database tables <u>and loaded mock data</u> saved as a .tar file.
5. SQL Queries that provide proof of normalization for basic business reporting queries. Create more than six SELECT queries to prove out normalization and table JOINs.