**Eggshell Database**

**User Guide**

**Table of Contents**

1. Database Environment Set Up
2. General Instructions
   1. Database Entities
   2. Database Objects
3. Instructions for Sales Representatives
4. Instructions for Construction Representatives

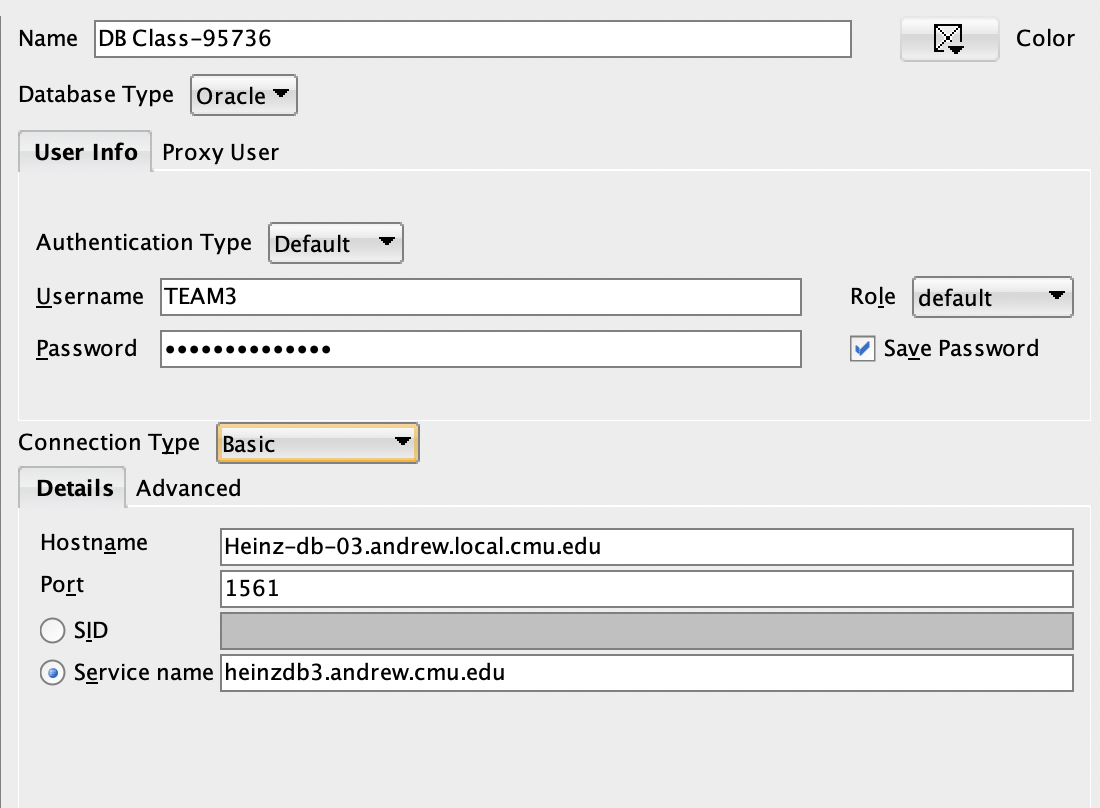
**1. Database Environment Set Up**

1. Log in

* Use Cisco AnyConnect Secure Mobility Client to get connected into the vpn.cmu.edu
* Click add new connection, and use the account to log in to this connection

Username: TEAM3

Password: HStud3nt\_Cla22

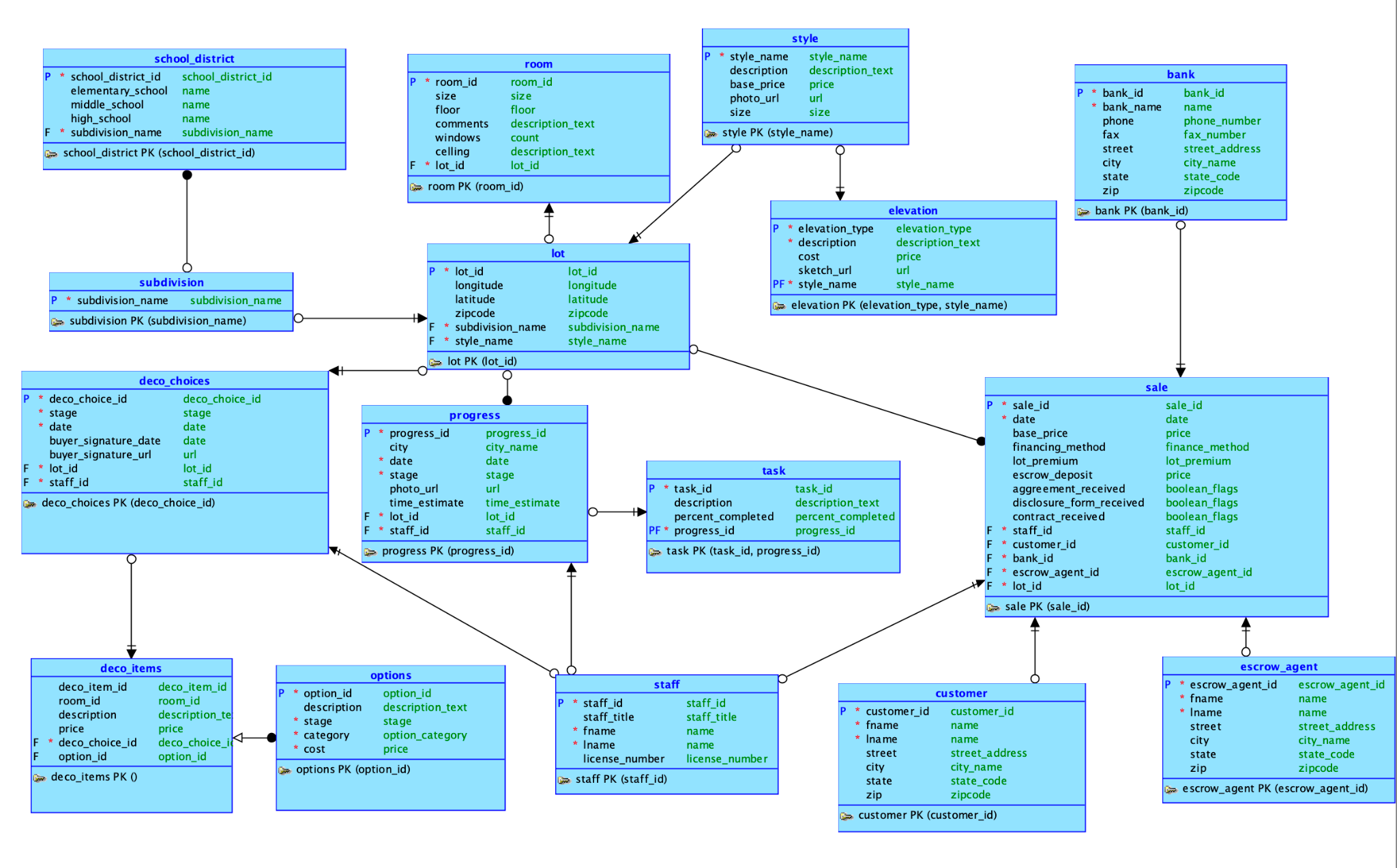
Also fill in the hostname, port, and service name as below

*2. Starting the script (which is usually not necessary)*

* *Unzip the data dictionary and view the basic information*
* *Open the SQLDeveloper, by typing ‘start [directory path]\G3\_eggshell.sql’, the tables will be created, and the views, triggers, sequences, procedures, packages, functions will be run*
* *Type: ‘SET SERVEROUTPUT ON’, before running the procedures, functions, and anything else to ensure the output will be shown*

**2. General Instructions (Database Overview)**

**2.1 Database Entities**



Before navigating the database, please read the entity description document carefully.

**2.2 Database Objects:**

**2.2.1 Sequences**

**Surrogate Key for Subdivision - subdivision\_sequence**

The subdivision sequence is created to facilitate the generation of the surrogate key for subdivisions in the subdivisions table. The table originally contained subdivision names as its primary key, thus a surrogate key is helpful for this table. This sequence starts at 1 and increments by 1. Existing entries of the table are assigned sequence upon creation of this sequence, and the future additions into the subdivision table will use the sequence to generate surrogate keys upon entry.

**Surrogate Key for Style - style\_sequence**

Similar to the subdivision table, the style sequence is created to facilitate the generation of the surrogate keys for the styles in the style table. The table originally contained style names as its primary key, thus a surrogate key is helpful for this table. This sequence starts at 1 and increments by 1. Existing entries of the table are assigned sequence upon creation of this sequence, and future additions into the subdivision table will use this sequence to generate surrogate keys upon entry.

**2.2.2 Views**

**View of near-finish properties - *near\_finish\_properties***

This is a view to help the company prepare for the delivery of construction properties. It lists the properties that are estimated to finish its construction in 30 days. To help the company arrange delivery and contact the relevant stakeholders, the table lists the names and information about the customer and the staff responsible for the property. The records are ordered in a descending order by remaining time estimates.

**Views of popular decoration options - *popular\_deco\_options***

This is a view to help the business analytics of the eggshell company. It aggregates and lists the decoration options popular among its customers, calculated by the number of times the option was selected by customers. The view contains information about the popular options’s id, name, description, category, and the stage it got chosen (so an option being chosen at a different stages are counted separately). The table helps stakeholders understand what options are popular, what are their characteristics, and during which stage do they get selected.

**2.2.3 Procedures**

**Update construction progress - *update\_construction\_progress***

The procedure is used to update construction progress for the company’s properties. This is executed when construction managers or staff check-in at construction sites and report back the newest statistics and information about the progress. The procedure prompts for the lot id to update, the new stage of the construction, and the new time estimate for the construction. Upon inputting those information, the procedure first checks if there is such a lot id undergoing construction. Then, it updates the construction stage and estimate of this lot in the progress table and enters the time stamp (date updated). If the estimated finish time is less than 5 days, the procedure then goes to all the tasks associated with this property to mark them as 100 percent completed, which helps us to focus on those ongoing tasks.

***Addition of subdivision - add\_subdivision***

This procedure is designed to work with the surrogate key subdivision\_id and the sequence subdivision sequence. When the procedure is executed, it simply prompts for the name of the new subdivision, it then increments the subdivision\_sequene and obtains the new surrogate key for the entry. It inserts the subdivision\_name and the surrogate subdivision\_id into the subdivision table.

**2.2.4 Function**

**Calculation of the total sales price - *calculate\_total\_sale***

This function calculates the total sales amount associated with each sale. For each sale, the function calculates the sum of base price and lot premium to output the total sales price.

**2.2.5 Package**

**The database package - *egg\_shell\_package***

This is the package used for the eggshell database, containing the two procedures (update\_construction\_progress and add\_subdivision) and one function (calculate\_total\_sale). Objects in this package are used for eggshell's daily operations.

**2.2.6 database triggers**

**Construction Progress Update - *log\_lot\_progress\_update***

This is a helper trigger to show and help log the updates on construction progress. Every time an update is made on construction progress, the trigger is executed to print out a message informing the user logging information, which include the lot number of interest, the original stage, and the new stage, as well as the time of the update.

**Subdivision Duplicate Check - no\_dup\_subdivision**

This trigger is executed when inserting new subdivisions into the subdivision table. If a subdivision with that name already exists, the trigger will raise an application error to warn the user the name exists.

**2.2.7 Scheduled Job**

**Yearly Price Raise for Inflation - *yearly\_price\_raise\_job***

This scheduled job is executed every 1 year. It raises the prices of all the decoration options by 2% every year to account for inflation. The job *yearly\_price\_raise\_job* is scheduled to execute the procedure called *yearly\_price\_raise* every year, which performs the corresponding actions to update price.

**2.2.8 Roles**

**Construction Manager - *construction\_manager\_role***

This is the role for construction managers.They are granted privileges on all tables associated with properties and construction progress, including property information, construction progress information, and tasks information. They can select and view tables related to different aspects and characteristics of the properties. They have additional rights to insert and update the construction progress and individual tasks to update relevant information.

| **Table** | ***SELECT*** | ***INSERT*** | ***UPDATE*** |
| --- | --- | --- | --- |
| **sale** | ***✔*** |  |  |
| **customer** | ***✔*** |  |  |
| **lot** | ***✔*** |  |  |
| **elevation** | ***✔*** |  |  |
| **style** | ***✔*** |  |  |
| **subdivision** | *✔* |  |  |
| **deco\_choices** | ***✔*** |  |  |
| **deco\_items** | ***✔*** |  |  |
| **options** | ***✔*** | ***✔*** | ***✔*** |
| **progress** | ***✔*** | ***✔*** | ***✔*** |
| **task** | ***✔*** | ***✔*** | ***✔*** |

**2.2.9 Denormalization**

**Denormalization of subdivision and school district tables**

When the sales representatives approach customers and tell them about the school district information of properties, with the use of surrogate id for subdivisions (which we implemented earlier), they will no longer have direct access to subdivision names in the school district table. When looking up school information about specific subdivisions, they will always have to join the two tables. This denormalization will thus speed up the look up process and improve query performance. The denormalization instance merged school information into subdivision tables and dropped the school district table.

**2.2.10 Alternate Index**

**Customer full name index - *customer\_name\_idx***

This alternate index is created for the combination of customer first name and customer last name. Given that sales agents often look up customers by their full name rather than remembering their customer id, creating this alternate index will speed up the look up process and thus improve database performance.

**3. Instructions for Sales Representatives**

**3.1 Role Privilege Description**

**Sales Agent - sales\_agent\_role**

This is the role for sales agents.They are granted privileges on all tables associated with sales, including property information, customer information, and bank information. They can select and view tables related to different aspects and characteristics of the properties. They have additional rights to insert and update the customer and sales table to update relevant information.

| **Table** | ***SELECT*** | ***INSERT*** | ***UPDATE*** |
| --- | --- | --- | --- |
| **sale** | ***✔*** | ***✔*** | *✔* |
| **bank** | ***✔*** |  |  |
| **customer** | ***✔*** | ***✔*** | ***✔*** |
| **lot** | ***✔*** |  |  |
| **elevation** | ***✔*** |  |  |
| **options** | *✔* |  |  |
| **style** | ***✔*** |  |  |
| **subdivision** | ***✔*** |  |  |
| **school\_district** | ***✔*** |  |  |
| **deco\_choices** | ***✔*** | ***✔*** | ***✔*** |
| **deco\_items** | ***✔*** | ***✔*** | ***✔*** |

\

**3.2 Decoration Choices Updating**

When buyers want to modify their decorator choices (eg. interior options) at different stages of the construction process (stages 1, 4, and 7), sales representativeshelp them to update the decorator items and thus update the decorator choice table(date of such modification). There might be a change in cost in labor and material associated with such modifications.

To insert or update a decorator choice, the responsible manager should create or update deco\_items. To do so, the manager first looks up the option id associated with the customer's choice (options are given in the options table, with set details and prices). To update deco\_item, managers add or replace deco\_item entry with the corresponding option\_id, price, description, room\_id, and the corresponding decorator\_choice\_id of the ticket. Eggshell will also assign such action with a deco\_item\_id. As business grows, eggshell will consider using sequence to populate deco\_item\_id. When customers choose the decorator items, they need to choose from the option tables. They can choose as many options as they want, as long as the options are available/applicable, and one deco\_item entry is created for each option applied.

**3.3 Updating Details of Sales**

While the base price is firm based on different lots, the lot premium is flexible based on different choices made by the customers and they can negotiate the price with sales representatives. Sales representatives have responsibility to set up and record accurate base prices and lot premiums.

To compute the total price (base price + premium) and to avoid errors and improve consistency, the sales manager must use the function given in the database calculate\_total\_sale.

Sales representatives and customers should be aware that eggshell's prices increase by 2 percent every year to account for inflation. This price increase procedure is a scheduled job in the database which will be executed every 1 year.

**3.4 Business Analytics for Sales Managers**

The options are constantly updated according to demands and business needs. To perform those business analytics, a view named popular\_deco\_options is created which presents popular options and stages of them being chosen. Sales managers should check the view often to update business and supply chain strategies.

**4. Instructions for Construction Representatives**

**4.1 Role Privilege Description**

**Construction Manager - construction\_manager\_role**

This is the role for construction managers.They are granted privileges on all tables associated with properties and construction progress, including property information, construction progress information, and tasks information. They can select and view tables related to different aspects and characteristics of the properties. They have additional rights to insert and update the construction progress and individual tasks to update relevant information.

| **Table** | ***SELECT*** | ***INSERT*** | ***UPDATE*** |
| --- | --- | --- | --- |
| **sale** | ***✔*** |  |  |
| **customer** | ***✔*** |  |  |
| **lot** | ***✔*** |  |  |
| **elevation** | ***✔*** |  |  |
| **style** | ***✔*** |  |  |
| **subdivision** | *✔* |  |  |
| **deco\_choices** | ***✔*** |  |  |
| **deco\_items** | ***✔*** |  |  |
| **options** | ***✔*** | ***✔*** | ***✔*** |
| **progress** | ***✔*** | ***✔*** | ***✔*** |
| **task** | ***✔*** | ***✔*** | ***✔*** |

**4.2 Construction Progress Updating**

**Construction manager** must update the construction progress for each lot, and include the latest stage and estimated completion time; **Construction manager** must be able to update buyers on the construction status.

When there are construction updates, the construction manager can choose to update the individual task items or update the construction progress directly. For example, if the manager chooses to update the task of fitting windows for a given property, they may choose to update the percentage completion of that specific task. If there are no specific task level updates, or if they simply want to give a broader overview update, they may choose to update the time estimate in the progress table. To update the progress table, the managers must execute the database procedure update\_construction\_progress.

To handle the potential discrepancies between individual tasks and the overall progress, if the estimated time for the overall progress is less than 5 days, all tasks associated with that progress will be marked complete (percentage complete to 100). This ensures that no tasks linger after its parent progress is completed.

When progress is updated in the database, the trigger log\_lot\_progress\_update will execute. In the console, oracle will print out a summary report of the change, which includes the lot id, old stage number, new stage number, and date changed. The manager must confirm and log those outputs to ensure consistency.

**4.3 Project Analysis for Construction Managers**

When a construction is about to finish, the construction managers are responsible for the handoff of the construction site to the sales managers and the clients. The planning of this handoff should start at least 30 days earlier than the estimated completion date. To see a list of near finish projects, construction managers should use the view near\_finish\_properties, which lists the names and information about the customer and the staff responsible for the property. The records are ordered in a descending order by remaining time estimates. Referring to this information, construction managers and other representatives should contact the other two parties (staff and customer) to arrange the handoff and delivery.