**Eggshell Database Object Description**

**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. 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.

**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.

**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.

**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.

**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.

**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.

**8. Roles**

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

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

\

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.

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

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

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.

**9. De-normalization**

* **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.

**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.