**Homework, Creating Tables and Views**

**All tables are in the sql subdirectory**

1. **Creating and Using a View**
	1. Create a view named **work.Phone\_List** containing the following columns:
* **Department** (**format=$25.**)
* **Name** (**format=$25.**)
* **Phone\_Number** (**label='Home Phone'** and **format=$16.**)

Use data found in the following tables: (Columns of interest are in parentheses.)

* **orion.Employee\_Addresses** (**Employee\_ID**, **Employee\_Name**)
* **orion.Employee\_Organization** (**Employee\_ID**, **Department**)
* **orion.Employee\_Phones** (**Employee\_ID**, **Phone\_Number**, **Phone\_Type**)

Include only those phone number records where **Phone\_Type="Home"**.

* 1. Use the new **work.Phone\_List** view as the source for a query that produces the phone list for the Engineering Department, sorted by **Name**. Add this title to the report: **Engineering Department Home Phone Numbers**.
1. **Creating and Using a View to Provide Consolidated Information**
	1. Create a view named **work.T\_Shirts** containing the following columns:
* **Product\_ID**
* **Supplier\_Name** (formatted as $20.)
* **Product\_Name**
* **Price** (Label it **Retail Price**.)

Use data from the following tables: (Columns of interest are in parentheses.)

* **orion.Product\_Dim** (**Product\_ID**, **Product\_Name**, **Supplier\_Name**)
* **orion.Price\_list** (**Product\_ID**, **Unit\_Sales\_Price**)

Select only those records where **Product\_Name** includes the word **T-Shirt**.

* 1. Write a query to display the data from your new **work.T\_Shirts** view. Sort the report by **Supplier\_Name** and **Product\_ID**. Supply a useful title.
	2. Write a query using your new **work.T\_Shirts** view to display **Product\_ID**, **Product\_Name**, and **Price** for all T-shirts priced less than $20.00. Sort the report by price. Supply a useful title.
1. **Creating and Using a View That Updates Itself over Time**
	1. Create a view named **work.Current\_Catalog**. The view must contain all of the columns in **orion.Product\_dim** and a new column named **Price** (labeled **Current Retail Price**). Use data found in the following tables:
* **orion.Product\_dim**
* **orion.Price\_list**

The value of the current price (**Price**) is determined by multiplying the original **Unit\_Sales\_Price** times the number of years since the product was first offered times the inflation **Factor**. Calculate the number of years that a product was offered by subtracting the year in which the product was first offered from the current year. The inflation **Factor** is a percentage of the original price. The formula is shown below:

**Price=Unit\_Sales\_Price \*
 (Factor\*\*(year(Today())-year(Start\_Date)))**

Round the calculated **Price** to the nearest cent.

* 1. Write a query to display **Supplier\_Name**, **Product\_Name**, and **Price** from your new **work.Current\_Catalog** view for products with “Roller Skate” in the product name. Sort the output by supplier name and price. Supply a useful title for your report. If the current year is later than 2009 your results will differ, as prices will have increased.
	2. Write a query to display **Product\_Name**, original **Unit\_Sales\_Price** (former **Price**), current **Price**, and the amount of **Increase** (calculated as **Price** − **Unit\_Sales\_Price**) for all products having an increase greater than $5.00. Sort the report by decreasing **Increase**. Supply a useful title. If the current year is later than 2009 your results will differ, as prices will have increased.
1. **Creating a Table and Adding Data Using a Query**
	1. Create a table containing the following columns and name it **work.Employees**:
* **Employee\_ID**
* **Hire\_Date**
* **Salary**
* **Birth\_Date**
* **Gender**
* **Country**
* **City**

Format all date columns with **MMDDYYYY10**. and **Salary** with **COMMA10.2**. Only include current employees (rows where **Employee\_Term\_Date** is missing). Order the output by year (**Employee\_Hire\_Date**) and then by descending **Salary**. The data that you need can be obtained from these tables:

* **orion.Employee\_Addresses**
* **orion.Employee\_Payroll**

Column Sourcing Information

|  |  |  |
| --- | --- | --- |
| **Employees** | **Employee\_Addresses** | **orion.Employee\_Payroll** |
| **Employee\_ID** | **Employee\_ID** | **Employee\_ID** |
| **Hire\_Date** | **-** | **Employee\_Hire\_Date** |
| **Salary** | **-** | **Salary** |
| **Birth\_Date** | **-** | **Birth\_Date** |
| **Gender** | **-** | **Employee\_Gender** |
| **Country** | **Country** | **-** |
| **City** | **City** | **-** |

* 1. Query the new table **work.Employees** to list all female employees whose salary is greater than $75,000.
1. **Creating a Table and Inserting Data Using a Complex Query**

Create a table named **Direct\_Compensation** in the Work library. The table should contain the following information for all **non-managerial** Sales staff (those with a level listed in their titles), formatted as indicated:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Employee\_ID** | **Name** | **Level** | **Salary** | **Commission** | **Direct\_Compensation** |
| 12345 | First Last | (I, II, III or IV) | 12,345.00  | 1,234.00  | 13,579.00 |

* Sales information is available in **orion.Order\_fact**.
* The table **orion.Sales** contains **Employee\_ID**, **First\_Name**, **Last\_Name**, **Job\_Title**, and **Salary** information for all Sales staff.
* **Job\_Title** contains level information for each employee.
* To calculate **Commission**, add the **Total\_Retail\_Price** values for all sales made by an employee with an **Order\_Date** in 2007, and take 15% of that total value as commission.
* To calculate **Direct\_Compensation**, add **Commission** plus **Salary**.

Add a query to list all of the data in the work.direct\_compensation table.