**HELP DESK SYSTEM**

DATABASE PROJECT REPORT

Ezgi GÖKDEMİR, Melike KINIŞ, Murat SÜNGÜ, Zeynep TOPÇU

JANUARY / 2021

**TABLES**

**Table : EMPLOYEE**

**The table keeps the information of the software company employees using the system.**

**Columns**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE | N | DESCRIPTION / ATTRIBUTES |
| PK | ID | int |  | Primary key for EMPLOYEE records Identity / Auto increment column |
|  | EMPLOYEE\_NAME | nvarchar(50) |  | Name of employee |
|  | EMPLOYEE\_SURNAME | nvarchar(50) |  | Surname of employee |
|  | MOBILE\_NUMBER | nvarchar(15) |  | Phone number of employee |
|  | EMPLOYEE\_EMAIL | nvarchar(50) |  | E-mail address of employee |
| FK | PROFESSIONID | int |  | Unique identification number for profession Foreign key to PROFESSION table |
| FK | DEPARTMENTID | int |  | Unique identification number for deparment  Foreign key to DEPARTMENT table |
|  | CREATE\_DATE | datetime |  | Create date of employee record |
|  | MODIFICATION\_DATE | datetime |  | Modification date of employee record |
|  | RECORD\_STATUS | bit |  | Status for employee record |
| FK | ROLEID | int |  | Unique identification number for role Foreign key to ROLE table |

**Links to**

|  |  |  |  |
| --- | --- | --- | --- |
| TABLE | | JOIN | TITLE / NAME / DESCRIPTION |
| PKC | PROFESSION | **EMPLOYEE**.PROFESSIONID =PROFESSION.ID | FK\_EMPLOYEE\_PROFESSION\_PROFESSIONID foreign key constraint referencing PROFESSION.ID |
| PKC | DEPARTMENT | **EMPLOYEE**.DEPARTMENTID = DEPARTMENT.ID | FK\_EMPLOYEE\_DEPARTMENT\_DEPARTMENTID foreign key constraint referencing DEPARTMENT.ID |
| PKC | ROLE | **EMPLOYEE**.ROLEID= ROLE.ID | FK\_EMPLOYEE\_ROLE\_ROLEID foreign key constraint referencing ROLE.ID |

**Linked from**

|  |  |  |  |
| --- | --- | --- | --- |
| TABLE | | JOIN | TITLE / NAME / DESCRIPTION |
| FKC | DEMAND | **EMPLOYEE**.ID =  DEMAND. EMPLOYEEID | FK\_DEMAND\_EMPLOYEE\_EMPLOYEEID foreign key constraint referencing EMPLOYEE.ID |
| FKC | PRODUCT\_EMPLOYEE\_MAPPING | **EMPLOYEE**. ID = PRODUCT\_EMPLOYEE\_MAPPING.EMPLOYEEID | FK\_PRODUCTEMPLOYEEMAPPING\_EMPLOYEE\_EMPLOYEEID foreign key constraint referencing EMPLOYEE.ID |

**Unique keys**

|  |  |  |
| --- | --- | --- |
| COLUMNS | | NAME / DESCRIPTION |
| PK | ID | PK\_EMPLOYEE\_ID Primary key (clustred) constraint |

**Uses**

|  |
| --- |
| NAME |
| **EMPLOYEE** |
| PROFESSION |
| DEPARMENT |
| ROLE |

**Used by**

|  |
| --- |
| NAME |
| **EMPLOYEE** |
| DEMAND |
| PRODUCT\_EMPLOYEE\_MAPPING |

**Table : PROFESSION**

**The table keeps job information for software company employees.**

**Columns**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE | N | DESCRIPTION / ATTRIBUTES |
| PK | ID | İnt |  | Primary key for PROFESSION records Identity / Auto increment column |
|  | PROFESSION\_NAME | nvarchar(50) |  | Name of profession |

**Linked from**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TABLE | | JOIN | | TITLE / NAME / DESCRIPTION |
| FKC | EMPLOYEE | | **PROFESSION**.ID = EMPLOYEE.PROFESSIONID | FK\_EMPLOYEE\_PROFESSION\_PROFESSIONID foreign key constraint referencing PROFESSION.ID |

**Unique keys**

|  |  |  |
| --- | --- | --- |
| COLUMNS | | NAME / DESCRIPTION |
| PK | ID | PK\_PROFESSION\_ID Primary key (clustred) constraint |

|  |  |  |  |
| --- | --- | --- | --- |
| **Used by** |  |  |  |
| NAME | | | |
| **PROFESSION** | | | |
| EMPLOYEE | | | |

**Table : DEPARTMENT**

**The table keeps department information of software company employees**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Columns** | |  |  |  |
| NAME | | DATA TYPE | N | DESCRIPTION / ATTRIBUTES |
| PK | ID | int |  | Primary key for DEPARTMENT records Identity / Auto increment column |
|  | DEPARTMENT\_NAME | nvarchar(50) |  | Name of department |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Linked from** | |  |  |  |
| TABLE | | JOIN | TITLE / NAME / DESCRIPTION | |
| FKC | EMPLOYEE | **DEPARTMENT**.ID = EMPLOYEE.DEPARTMENTID | FK\_EMPLOYEE\_DEPARTMENT\_DEPARMENTID foreign key constraint referencing DEPARTMENT.ID | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Unique keys** | |  |  |  |
| COLUMNS | | NAME / DESCRIPTION | | |
| PK | ID | PK\_DEPARTMENT\_ID Primary key (clustred) constraint | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Used By** |  |  |  |
| NAME | | | |
| **DEPARTMENT** | | | |
| EMPLOYEE | | | |

**Table : ROLE**

**The table keeps information about the roles of the actors in the system.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Columns** | |  |  |  |  |
| NAME | | | DATA TYPE | N | DESCRIPTION / ATTRIBUTES |
| PK | ID | | İnt |  | Primary key for ROLE records Identity / Auto increment column |
|  | ROLE\_NAME | | nvarchar(30) |  | Name of role |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Linked from** | | |  | |  | |  |  |
|  | TABLE | | | | | JOIN | | TITLE / NAME / DESCRIPTION | |
|  | FKC | | COMPANY\_USER | | | **ROLE**.ID = COMPANY\_USER.ROLEID | | FK\_COMPANYUSER\_ROLE\_ROLEID foreign key constraint referencing ROLE.ID | |
|  | FKC | | EMPLOYEE | | | **ROLE**.ID = EMPLOYEE.ROLEID | | FK\_EMPLOYEE\_ROLE\_ROLEID foreign key constraint referencing ROLE.ID | |
| **Unique keys** | | | | |  | |  | | |  | |  |
| COLUMNS | | | | | | | NAME / DESCRIPTION | | | |
| PK | | ID | | | | | PK\_ROLE\_ID Primary key (clustred) constraint | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Used by** |  |  |  |  |
| NAME | | | | |
| **ROLE** | | | | |
| COMPANY\_USER\_ROLE\_MAPPING | | | | |
| EMPLOYEE\_ROLE\_MAPPING | | | | |

**Table : PRODUCT\_EMPLOYEE\_MAPPING**

**The table keeps the information that which employee can be assigned to the demand about which product.**

**Columns**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE | N | DESCRIPTION / ATTRIBUTES |
| PK | ID | int |  | Primary key for PRODUCT\_EMPLOYEE\_MAPPING records Identity / Auto increment column |
| FK | PRODUCTID | int |  | Unique identification number for product Foreign key to PRODUCT table |
| FK | EMPLOYEEID | int |  | Unique identification number for employee Foreign key to EMPLOYEE table |
|  | RECORD\_STATUS | bit |  | Status for product employee mapping record |

**Links to**

|  |  |  |  |
| --- | --- | --- | --- |
| TABLE | | JOIN | TITLE / NAME / DESCRIPTION |
| PKC | PRODUCT | **PRODUCT\_EMPLOYEE\_MAPPING**.PRODUCTID= PRODUCT.ID | FK\_PRODUCTEMPLOYEEMAPPING\_PRODUCT\_PRODUCTID foreign key constraint referencing PRODUCT.ID |
| PKC | EMPLOYEE | **PRODUCT\_EMPLOYEE\_MAPPING**.EMPLOYEEID= EMPLOYEE.ID | FK\_PRODUCTEMPLOYEEMAPPING\_EMPLOYEE\_EMPLOYEEID foreign key constraint referencing EMPLOYEE.ID |

**Unique keys**

|  |  |  |
| --- | --- | --- |
| COLUMNS | | NAME / DESCRIPTION |
| PK | ID | PK\_PRODUCT\_EMPLOYEE\_MAPPING \_ID Primary key (clustred) constraint |

**Uses**

|  |
| --- |
| NAME |
| **COMPANY\_PRODUCT\_MAPPING** |
| PRODUCT |
| EMPLOYEE |

**Table : COMPANY\_USER**

**The table keeps information about the people who use the product of software company.**

**Columns**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE | N | DESCRIPTION / ATTRIBUTES |
| PK | ID | İnt |  | Primary key for COMPANY\_USER records Identity / Auto increment column |
|  | COMPANY\_USER\_NAME | nvarchar(50) |  | Name of company user |
|  | COMPANY\_USER\_SURNAME | nvarchar(50) |  | Surname of company user |
|  | MOBILE\_NUMBER | nvarchar(15) |  | Phone number of company user |
|  | COMPANY\_USER\_EMAIL | nvarchar(50) |  | E-mail address of company user |
| FK | COMPANYID | İnt |  | Unique identification number for company. Foreign key to COMPANY table |
|  | CREATE\_DATE | Datetime |  | Create date of company user record |
|  | MODIFICATION\_DATE | Datetime |  | Modification date of company user record |
|  | RECORD\_STATUS | Bit |  | Status for company user record |
| FK | ROLEID | İnt |  | Unique identification number for role Foreign key to ROLE table |

**Links to**

|  |  |  |  |
| --- | --- | --- | --- |
| TABLE | | JOIN | TITLE / NAME / DESCRIPTION |
| PKC | COMPANY | **COMPANY\_USER**.COMPANYID= COMPANY.ID | FK\_COMPANYUSER\_COMPANY\_COMPANYID foreign key constraint referencing COMPANY.ID |
| PKC | ROLE | COMPANY\_USER.ROLEID= ROLE.ID | FK\_COMPANYUSER\_ROLE\_ROLEID foreign key constraint referencing ROLE.ID |

**Linked from**

|  |  |  |  |
| --- | --- | --- | --- |
| TABLE | | JOIN | TITLE / NAME / DESCRIPTION |
| FKC | DEMAND | **COMPANY\_USER**.ID = DEMAND. COMPANY\_USERID | FK\_DEMAND\_COMPANY\_USER\_COMPANY\_USERID foreign key constraint referencing COMPANY\_USER.ID |

**Unique keys**

|  |  |  |
| --- | --- | --- |
| COLUMNS | | NAME / DESCRIPTION |
| PK | ID | PK\_COMPANY\_USER\_ID Primary key (clustred) constraint |

**Uses**

|  |
| --- |
| NAME |
| **COMPANY\_USER** |
| COMPANY |
| ROLE |

**Used by**

|  |
| --- |
| NAME |
| **COMPANY\_USER** |
| DEMAND |

**Table : COMPANY**

**The table keeps information about companies that are customers of the software company.**

**Columns**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE | N | DESCRIPTION / ATTRIBUTES |
| PK | ID | İnt |  | Primary key for COMPANY records Identity / Auto increment column |
|  | COMPANY\_NAME | nvarchar(255) |  | Name of company |
|  | WEBSITE | nvarchar(100) |  | Website of company |
|  | COMPANY\_EMAIL | nvarchar(100) |  | E-mail address of company |
|  | FIXED\_NUMBER | nvarchar(15) |  | Phone number of company |
|  | RECORD\_STATUS | Bit |  | Status for company record |

**Linked from**

|  |  |  |  |
| --- | --- | --- | --- |
| TABLE | | JOIN | TITLE / NAME / DESCRIPTION |
| FKC | COMPANY\_USER | **COMPANY.ID** = COMPANY\_USER.COMPANYID | FK\_COMPANYUSER\_COMPANY\_COMPANYID foreign key constraint referencing COMPANY.ID |
| FKC | COMPANY\_ADDRESS\_MAPPING | **COMPANY.ID** = COMPANY\_ADDRESS\_MAPPING. COMPANYID | FK\_ADDRESS\_COMPANY\_COMPANYID foreign key constraint referencing COMPANY.ID |
| FKC | COMPANY\_PRODUCT\_MAPPING | **COMPANY.ID**= COMPANY\_PRODUCT\_MAPPING.COMPANYID | FK\_COMPANYPRODUCTMAPPING\_COMPANY\_COMPANYID foreign key constraint referencing COMPANY.ID |

**Unique keys**

|  |  |  |
| --- | --- | --- |
| COLUMNS | | NAME / DESCRIPTION |
| PK | ID | PK\_COMPANY\_ID Primary key (clustred) constraint |

**Used by**

|  |
| --- |
| NAME |
| **COMPANY** |
| COMPANY\_USER |
| COMPANY\_ADDRESS\_MAPPING |
| COMPANY\_PRODUCT\_MAPPING |

**Table : COMPANY\_ADDRESS\_MAPPING**

**A company can have more than one address. The table keeps the address information of the companies registered in the system.**

**Columns**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE | N | DESCRIPTION / ATTRIBUTES |
| PK | ID | İnt |  | Primary key for COMPANY\_ADDRESS\_MAPPING records Identity / Auto increment column |
| FK | COMPANYID | İnt |  | Unique identification number for company  Foreign key to COMPANY table |
| FK | PROVINCEID | İnt |  | Unique identification number for province Foreign key to PROVINCE table |
| FK | DISTRICTID | İnt |  | Unique identification number for district Foreign key to DISTRICT table |
|  | ADDRESS\_LINE | nvarchar(250) |  | Full address of the company |

**Links to**

|  |  |  |  |
| --- | --- | --- | --- |
| TABLE | | JOIN | TITLE / NAME / DESCRIPTION |
| PKC | COMPANY | **COMPANY\_ADDRESS\_MAPPING**.COMPANYID =COMPANY.ID | FK\_ADDRESS\_COMPANY\_COMPANYID foreign key constraint referencing COMPANY.ID |
| PKC | PROVINCE | **COMPANY\_ADDRESS\_MAPPING**.PROVINCEID = PROVINCE.ID | FK\_ADDRESS\_PROVINCE\_PROVINCEID foreign key constraint referencing PROVINCE.ID |
| PKC | DISTRICT | **COMPANY\_ADDRESS\_MAPPING.**DISTRICTID = DISTRICT.ID | FK\_ADDRESS\_DISTRICT\_DISTRICTID foreign key constraint referencing DISTRICT.ID |

**Unique keys**

|  |  |  |
| --- | --- | --- |
| COLUMNS | | NAME / DESCRIPTION |
| PK | ID | PK\_COMPANY\_ADDRESS\_MAPPING\_ID Primary key (clustred) constraint |

**Uses**

|  |
| --- |
| NAME |
| **COMPANY\_ADDRESS\_MAPPING** |
| COMPANY |
| PROVINCE |
| DISTRICT |

**Table : PROVINCE**

**The table keeps the city information of the address.**

**Columns**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE | N | DESCRIPTION / ATTRIBUTES |
| PK | ID | İnt |  | Primary key for PROVINCE records Identity / Auto increment column |
|  | PROVINCE\_NAME | nvarchar(30) |  | Name of province |

**Linked from**

|  |  |  |  |
| --- | --- | --- | --- |
| TABLE | | JOIN | TITLE / NAME / DESCRIPTION |
| FKC | COMPANY\_ADDRESS\_MAPPING | **PROVINCE**.ID = COMPANY\_ADDRESS\_MAPPING.PROVINCEID | FK\_ADDRESS\_PROVINCE\_PROVINCEID foreign key constraint referencing PROVINCE.ID |
| FKC | DISTRICT | **PROVINCE**.ID = DISTRICT.PROVINCEID | FK\_DISTRICT\_PROVINCE\_PROVINCEID foreign key constraint referencing PROVINCE.ID |

**Unique keys**

|  |  |  |
| --- | --- | --- |
| COLUMNS | | NAME / DESCRIPTION |
| PK | ID | PK\_PROVINCE\_ID Primary key (clustred) constraint |

**Used by**

|  |
| --- |
| NAME |
| **PROVINCE** |
| COMPANY\_ADDRESS\_MAPPING |
| DISTRICT |

**Table : DISTRICT**

**The table keeps the district information of the address.**

**Columns**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE | N | DESCRIPTION / ATTRIBUTES |
| PK | ID | İnt |  | Primary key for DISTRICT records Identity / Auto increment column |
|  | DISTRICT\_NAME | nvarchar(50) |  | Name of district |
| FK | PROVINCEID | İnt |  | Unique identification number for province. Foreign key to PROVINCE table |

**Links to**

|  |  |  |  |
| --- | --- | --- | --- |
| TABLE | | JOIN | TITLE / NAME / DESCRIPTION |
| PKC | PROVINCE | **DISTRICT**.PROVINCEID = PROVINCE.ID | FK\_DISTRICT\_PROVINCE\_PROVINCEID foreign key constraint referencing PROVINCE.ID |

**Linked from**

|  |  |  |  |
| --- | --- | --- | --- |
| TABLE | | JOIN | TITLE / NAME / DESCRIPTION |
| FKC | COMPANY\_ADDRESS\_MAPPING | **DISTRICT**.ID = COMPANY\_ADDRESS\_MAPPING.DISTRICTID | FK\_ADDRESS\_DISTRICT\_DISTRICTID foreign key constraint referencing DISTRICTID |

**Unique keys**

|  |  |  |  |
| --- | --- | --- | --- |
| COLUMNS | | NAME / DESCRIPTION | |
| PK | ID | | PK\_DISTRICT\_ID Primary key (clustred) constraint |

**Uses**

|  |
| --- |
| NAME |
| **DISTRICT** |
| PROVINCE |

**Used by**

|  |
| --- |
| NAME |
| **DISTRICT** |
| COMPANY\_ADDRESS\_MAPPING |

**Table : COMPANY\_PRODUCT\_MAPPING**

**The table keeps the information that which company buys which product.**

**Columns**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE | N | DESCRIPTION / ATTRIBUTES |
| PK | ID | İnt |  | Primary key for COMPANY\_PRODUCT\_MAPPING records Identity / Auto increment column |
| FK | PRODUCTID | İnt |  | Unique identification number for product Foreign key to PRODUCT table |
| FK | COMPANYID | İnt |  | Unique identification number for company Foreign key to COMPANY table |
|  | LICENSE\_PERIOD\_END\_DATE | Datetime |  | License ending date of product |
|  | CREATE\_DATE | Datetime |  | Create date of company product mapping record |
|  | RECORD\_STATUS | Bit |  | Status for company product mapping record |

**Links to**

|  |  |  |  |
| --- | --- | --- | --- |
| TABLE | | JOIN | TITLE / NAME / DESCRIPTION |
| PKC | PRODUCT | **COMPANY\_PRODUCT\_MAPPING**.PRODUCTID =  PRODUCT.ID | FK\_COMPANYPRODUCTMAPPING\_PRODUCT\_PRODUCTID foreign key constraint referencing PRODUCT.ID |
| PKC | COMPANY | **COMPANY\_PRODUCT\_MAPPING**.COMPANYID = COMPANY.ID | FK\_COMPANYPRODUCTMAPPING\_COMPANY\_COMPANYID foreign key constraint referencing COMPANY.ID |

**Unique keys**

|  |  |  |
| --- | --- | --- |
| COLUMNS | | NAME / DESCRIPTION |
| PK | ID | PK\_COMPANY\_PRODUCT\_MAPPING\_ID Primary key (clustred) constraint |

**Uses**

|  |
| --- |
| NAME |
| **COMPANY\_PRODUCT\_MAPPING** |
| PRODUCT |
| COMPANY |

**Table : PRODUCT**

**The table keeps information about the products of software company**

**Columns**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE | N | DESCRIPTION / ATTRIBUTES |
| PK | ID | İnt |  | Primary key for PRODUCT records Identity / Auto increment column |
|  | PRODUCT\_NAME | nvarchar(100) |  | Name of product |
| FK | LICENCE\_PERIODID | İnt |  | Unique identification number for license period Foreign key to LICENSE\_PERIOD table |

**Links to**

|  |  |  |  |
| --- | --- | --- | --- |
| TABLE | | JOIN | TITLE / NAME / DESCRIPTION |
| PKC | LICENCE\_PERIOD | **PRODUCT**.LICENCE\_PERIODID = LICENSE\_PERIOD.ID | FK\_PRODUCT\_LICENSE\_PERIOD\_LICENSE\_PERIODID foreign key constraint referencing LICENSE\_PERIOD.ID |

**Linked from**

|  |  |  |  |
| --- | --- | --- | --- |
| TABLE | | JOIN | TITLE / NAME / DESCRIPTION |
| FKC | PRODUCT\_EMPLOYEE\_MAPPING | **PRODUCT**.ID = PRODUCT\_EMPLOYEE\_MAPPING.PRODUCTID | FK\_PRODUCTROLEMAPPING\_PRODUCT\_PRODUCTID foreign key constraint referencing PRODUCT.ID |
| FKC | COMPANY\_PRODUCT\_MAPPING | **PRODUCT.**ID = COMPANY\_PRODUCT\_MAPPING. PRODUCTID | FK\_COMPANYPRODUCTMAPPING\_PRODUCT\_PRODUCTID foreign key constraint referencing PRODUCT.ID |
| FKC | DEMAND | **PRODUCT**.ID = DEMAND.PRODUCTID | FK\_DEMAND\_PRODUCT\_PRODUCTID foreign key constraint referencing PRODUCT.ID |
| FKS | VERSION | **PRODUCT**.ID = VERSION.PRODUCTID | FK\_VERSION\_PRODUCT\_PRODUCTID foreign key constraint referencing PRODUCT.ID |

**Unique keys**

|  |  |  |
| --- | --- | --- |
| COLUMNS | | NAME / DESCRIPTION |
| PK | ID | PK\_PRODUCT\_ID Primary key (clustred) constraint |

**Uses**

|  |
| --- |
| NAME |
| **PRODUCT** |
| LICENCE\_PERIOD |

**Used by**

|  |
| --- |
| NAME |
| **PRODUCT** |
| PRODUCT\_EMPLOYEE\_MAPPING |
| COMPANY\_PRODUCT\_MAPPING |
| DEMAND |
| VERSION |

**Table : LICENSE\_PERIOD**

**The table keeps information about license period of products**

**Columns**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE | N | DESCRIPTION / ATTRIBUTES |
| PK | ID | int |  | Primary key for LICENSE\_PERIOD records Identity / Auto increment column |
|  | PERIOD | nvarchar(50) |  | Period of license |

**Linked from**

|  |  |  |  |
| --- | --- | --- | --- |
| TABLE | | JOIN | TITLE / NAME / DESCRIPTION |
| FKC | PRODUCT | **LICENSE\_PERIOD**.ID = PRODUCT.LICENSE\_PERIODID | FK\_PRODUCT\_LICENSE\_PERIOD\_LICENSE\_PERIODID foreign key constraint referencing LICENSE\_PERIOD.ID |

**Unique keys**

|  |  |  |
| --- | --- | --- |
| COLUMNS | | NAME / DESCRIPTION |
| PK | ID | PK\_LICENSE\_PERIOD\_ID Primary key (clustred) constraint |

**Used by**

|  |
| --- |
| NAME |
| **LICENSE\_PERIOD** |
| PRODUCT |

**Table : VERSION**

**The table keeps the version information of the software products.**

**Columns**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE | N | DESCRIPTION / ATTRIBUTES |
| PK | ID | İnt |  | Primary key for VERSION records Identity / Auto increment column |
| FK | PRODUCTID | İnt |  | Unique identification number for product Foreign key to PRODUCT table |
|  | VERSIONNO | nvarchar(20) |  | Number of version |
|  | CREATE\_DATE | Datetime |  | Create date of version record |
|  | RECORD\_STATUS | Bit |  | Status for version record |

**Links to**

|  |  |  |  |
| --- | --- | --- | --- |
| TABLE | | JOIN | TITLE / NAME / DESCRIPTION |
| PKC | PRODUCT | **VERSION**.PRODUCTID= PRODUCT.ID | FK\_VERSION\_PRODUCT\_PRODUCTID foreign key constraint referencing PRODUCT.ID |

**Linked from**

|  |  |  |  |
| --- | --- | --- | --- |
| TABLE | | JOIN | TITLE / NAME / DESCRIPTION |
| FKC | DEMAND | **VERSION**.ID = DEMAND. VERSIONID | FK\_DEMAND\_VERSION\_VERSIONID foreign key constraint referencing VERSION.ID |

**Unique keys**

|  |  |  |
| --- | --- | --- |
| COLUMNS | | NAME / DESCRIPTION |
| PK | ID | PK\_VERSION\_ID Primary key (clustred) constraint |

**Uses**

|  |
| --- |
| NAME |
| **VERSION** |
| PRODUCT |

**Used by**

|  |
| --- |
| NAME |
| **VERSION** |
| DEMAND |

**Table : DEMAND**

**The table keeps information about demand created.**

**Columns**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE | N | DESCRIPTION / ATTRIBUTES |
| PK | ID | int |  | Primary key for DEMAND records Identity / Auto increment column |
|  | TITLE | nvarchar(max) |  | Subject of demand |
|  | TEXT | nvarchar(max) |  | Detailed explanation of demand |
| FK | DEMAND\_TYPEID | int |  | Unique identification number for demand type  Foreign key to DEMAND\_TYPE table |
| FK | ORDER\_OF\_URGENCYID | int |  | Unique identification number for order of urgency Foreign key to ORDER\_OF\_URGENCY table |
| FK | DEMAND\_STATEID | int |  | Unique identification number for order of demand state Foreign key to DEMAND\_STATE table |
|  | CLOSING\_STATEMENT | nvarchar(max) |  | Closing statement of demand |
|  | CLOSING\_DATE |  |  | Closing date of demand |
| FK | COMPANY\_USERID | int |  | Unique identification number for company user Foreign key to COMPANY\_USER table |
| FK | EMPLOYEEID | int |  | Unique identification number for employee Foreign key to EMPLOYEE table |
| FK | PRODUCTID | int |  | Unique identification number for product Foreign key to PRODUCT table |
|  | SOLVED\_HOUR | int |  | Closing time of demand /as hour |
| FK | VERSIONID | int |  | Unique identification number for version Foreign key to VERSION table |
|  | CREATE\_DATE | datetime |  | Create date of demand record |
|  | MODIFICATION\_DATE | datetime |  | Modification date of demand record |
|  | RECORD\_STATUS | bit |  | Status for demand record |

**Links to**

|  |  |  |  |
| --- | --- | --- | --- |
| TABLE | | JOIN | TITLE / NAME / DESCRIPTION |
| PKC | DEMAND\_TYPE | **DEMAND**.DEMAND\_TYPEID = DEMAND\_TYPE.ID | FK\_DEMAND\_DEMAND\_TYPE\_DEMAND\_TYPEID foreign key constraint referencing DEMAND\_TYPE.ID |
| PKC | ORDER\_OF\_URGENCY | **DEMAND**.ORDER\_OF\_URGENCYID = ORDER\_OF\_URGENCY.ID | FK\_DEMAND\_ORDER\_OF\_URGENCY\_ORDER\_OF\_URGENCYID foreign key constraint referencing ORDER\_OF\_URGENCY.ID |
| PKC | DEMAND\_STATE | **DEMAND**.DEMAND\_STATEID = DEMAND\_STATE.ID | FK\_DEMAND\_DEMAND\_STATE\_DEMAND\_STATEID foreign key constraint referencing DEMAND\_STATE.ID |
| PKC | COMPANY\_USER | **DEMAND**.COMPANY\_USERID = COMPANY\_USER.ID | FK\_DEMAND\_COMPANY\_USER\_COMPANY\_USERID foreign key constraint referencing COMPANY\_USER.ID |
| PKC | EMPLOYEE | **DEMAND**.EMPLOYEEID = EMPLOYEE.ID | FK\_DEMAND\_EMPLOYEE\_EMPLOYEEID foreign key constraint referencing EMPLOYEE.ID |
| PKC | PRODUCT | **DEMAND**.PRODUCTID = PRODUCT.ID | FK\_DEMAND\_PRODUCT\_PRODUCTID foreign key constraint referencing PRODUCT.ID |
| PKC | VERSION | **DEMAND**.VERSIONID= VERSION.ID | FK\_DEMAND\_VERSION\_VERSIONID foreign key constraint referencing VERSION.ID |

**Unique keys**

|  |  |  |  |
| --- | --- | --- | --- |
| COLUMNS | | | NAME / DESCRIPTION |
| PK | ID | PK\_DEMAND\_ID Primary key (clustred) constraint | |

**Uses**

|  |
| --- |
| NAME |
| **DEMAND** |
| DEMAND\_TYPE |
| ORDER\_OF\_URGENCY |
| DEMAND\_STATE |
| COMPANY\_USER |
| EMPLOYEE |
| PRODUCT |
| VERSION |

**Table : DEMAND \_TYPE**

**The table keeps information about the type of demand generated by the company user.**

**Columns**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE | N | DESCRIPTION / ATTRIBUTES |
| PK | ID | int |  | Primary key for DEMAND\_TYPE records Identity / Auto increment column |
|  | TYPE | nvarchar(20) |  | Type of demand |

**Linked from**

|  |  |  |  |
| --- | --- | --- | --- |
| TABLE | | JOIN | TITLE / NAME / DESCRIPTION |
| FKC | DEMAND | **DEMAND\_TYPE**.ID = DEMAND.DEMAND\_TYPEID | FK\_DEMAND\_DEMAND\_TYPE\_DEMAND\_TYPEID foreign key constraint referencing DEMAND\_TYPE.ID |

**Unique keys**

|  |  |  |
| --- | --- | --- |
| COLUMNS | | NAME / DESCRIPTION |
| PK | ID | PK\_DEMAND\_TYPE\_ID Primary key (clustred) constraint |

**Used by**

|  |
| --- |
| NAME |
| **DEMAND\_TYPE** |
| DEMAND |

**Table : ORDER\_OF\_URGENCY**

**The table keeps information on the urgency of the demand.**

**Columns**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE | N | DESCRIPTION / ATTRIBUTES |
| PK | ID | int |  | Primary key for ORDER\_OF\_URGENCY records Identity / Auto increment column |
|  | URGENCY | nvarchar(20) |  | Urgency of the demand |

**Linked from**

|  |  |  |  |
| --- | --- | --- | --- |
| TABLE | | JOIN | TITLE / NAME / DESCRIPTION |
| FKC | DEMAND | **ORDER\_OF\_URGENCY**.ID = DEMAND.ORDER\_OF\_URGENCYID | FK\_DEMAND\_ORDER\_OF\_URGENCY\_ORDER\_OF\_URGENCYID foreign key constraint referencing ORDER\_OF\_URGENCY.ID |

**Unique keys**

|  |  |  |
| --- | --- | --- |
| COLUMNS | | NAME / DESCRIPTION |
| PK | ID | PK\_ORDER\_OF\_URGENCY\_ID Primary key (clustred) constraint |

**Used by**

|  |
| --- |
| NAME |
| **ORDER\_OF\_URGENCY** |
| DEMAND |

**Table : DEMAND\_STATE**

**The table keeps information about the status of the demand.**

**Columns**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE | N | DESCRIPTION / ATTRIBUTES |
| PK | ID | int |  | Primary key for DEMAND\_STATE records Identity / Auto increment column |
|  | STATE | nvarchar(15) |  | State of demand |

**Linked from**

|  |  |  |  |
| --- | --- | --- | --- |
| TABLE | | JOIN | TITLE / NAME / DESCRIPTION |
| FKC | DEMAND | **DEMAND\_STATE**.ID = DEMAND.DEMAND\_STATEID | FK\_DEMAND\_DEMAND\_STATE\_DEMAND\_STATEID foreign key constraint referencing DEMAND\_STATE.ID |

**Unique keys**

|  |  |  |
| --- | --- | --- |
| COLUMNS | | NAME / DESCRIPTION |
| PK | ID | PK\_DEMAND\_STATE\_ID Primary key (clustred) constraint |

**Used by**

|  |
| --- |
| NAME |
| **DEMAND\_STATE** |
| DEMAND |

**FUNCTIONS**

**Function: f\_GET\_PRODUCTS\_FOR\_EMPLOYEE\_ID**

**Table-valued function returning the product name for the given employee id.**

**Input/Output**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE |  | DESCRIPTION / ATTRIBUTES |
| @ | RETURNS | TABLE TYPE |  | |
| @ | EMPLOYEEID | INT | Input parameter for the table-valued function f\_GET\_PRODUCTS\_BY\_EMPLOYEE\_ID. Enter a valid EMPLOYEEID from the EMPLOYEE table. | |

**Function: f\_GET\_DEMANDS\_FOR\_COMPANY\_USERID**

**Table-valued function returning the all columns of demand table  for the given company user id.**

**Input/Output**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE |  | DESCRIPTION / ATTRIBUTES |
| @ | RETURNS | TABLE TYPE |  | |
| @ | COMPANY\_USERID | INT | Input parameter for the table-valued function f\_GET\_DEMANDS\_BY\_COMPANY\_USERID. Enter a valid COMPANY\_USERID from the COMPANY\_USER table. | |

**Function: f\_GET\_ALL\_DEMAND\_URGENCY\_FOR\_EMPLOYEEID**

**Table-valued function returning the urgency types and total number of demands on the basis of  urgency type for the given employee id.**

**Input/Output**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE |  | DESCRIPTION / ATTRIBUTES |
| @ | RETURNS | TABLE TYPE |  | |
| @ | EMPLOYEEID | INT | Input parameter for the table-valued function f\_GET\_ALL\_DEMAND\_URGENCY\_BY\_EMPLOYEEID. Enter a valid EMPLOYEEID from the EMPLOYEE table. | |

**Function: f\_GET\_ALL\_DEMAND\_TYPES\_FOR\_EMPLOYEEID**

**Table-valued function returning the demand types and total number of demands on the basis of demand type for the given employee id.**

**Input/Output**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE |  | DESCRIPTION / ATTRIBUTES |
| @ | RETURNS | TABLE TYPE |  | |
| @ | EMPLOYEEID | INT | Input parameter for the table-valued function f\_GET\_ALL\_DEMAND\_TYPES\_BY\_EMPLOYEEID. Enter a valid EMPLOYEEID from the EMPLOYEE table. | |

**Function: f\_GET\_ALL\_DEMAND\_STATES\_FOR\_EMPLOYEEID**

**Table-valued function returning the demand states and total number of demands on the basis of demand state for the given employee id.**

**Input/Output**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE |  | DESCRIPTION / ATTRIBUTES |
| @ | RETURNS | TABLE TYPE |  | |
| @ | EMPLOYEEID | INT | Input parameter for the table-valued function f\_GET\_ALL\_DEMAND\_STATES\_BY\_EMPLOYEEID. Enter a valid EMPLOYEEID from the EMPLOYEE table. | |

**Function: f\_GET\_ALL\_DEMAND\_FOR\_COMPANY**

**Table-valued function returning the all columns of demand table for the given company id.**

**Input/Output**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE |  | DESCRIPTION / ATTRIBUTES |
| @ | RETURNS | TABLE TYPE |  | |
| @ | COMPANYID | INT | Input parameter for the table-valued function f\_GET\_ALL\_DEMAND\_FOR\_COMPANY. Enter a valid COMPANYID from the COMPANY table. | |

**Function: f\_COUNT\_OF\_DEMAND\_FOR\_COMPANY\_BY\_STATE**

**Table-valued function returning the demand state and count of demand on the basis of demand state for the given company id.**

**Input/Output**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE |  | DESCRIPTION / ATTRIBUTES |
| @ | RETURNS | TABLE TYPE |  | |
| @ | COMPANYID | INT | Input parameter for the table-valued function f\_COUNT\_OF\_DEMAND\_FOR\_COMPANY\_BY\_STATE. Enter a valid COMPANYID from the COMPANY table. | |

**Function: f\_GET\_ALL\_DEMAND\_FOR\_EMPLOYEEID**

**Table-valued function returning the demands for the given employee id.**

**Input/Output**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE |  | DESCRIPTION / ATTRIBUTES |
| @ | RETURNS | TABLE TYPE |  | |
| @ | EMPLOYEEID | INT | Input parameter for the table-valued function f\_GET\_ALL\_DEMAND\_BY\_EMPLOYEEID. Enter a valid EMPLOYEEID from the EMPLOYEE table. | |

**Function: f\_GET\_ALL\_DEMAND\_AND\_STATES\_FOR\_EMPLOYEEID**

**Table-valued function returning the all demands and demand states for the given employee id.**

**Input/Output**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE |  | DESCRIPTION / ATTRIBUTES |
| @ | RETURNS | TABLE TYPE |  | |
| @ | EMPLOYEEID | INT | Input parameter for the table-valued function f\_GET\_ALL\_DEMAND\_AND\_STATES\_BY\_EMPLOYEEID. Enter a valid EMPLOYEEID from the EMPLOYEE table. | |

**Function: f\_COUNT\_OF\_DEMAND\_FOR\_COMPANY\_USERID\_BY\_STATE**

**Table-valued function returning the demand state and count of demand on the basis of demand state for the given company user id.**

**Input/Output**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE |  | DESCRIPTION / ATTRIBUTES |
| @ | RETURNS | TABLE TYPE |  | |
| @ | COMPANY\_USERID | INT | Input parameter for the table-valued function f\_COUNT\_OF\_DEMAND\_FOR\_COMPANY\_USERID\_BY\_STATE. Enter a valid COMPANY\_USERID from the COMPANY\_USER table. | |

**Function: f\_GET\_DEMAND\_BY\_COMPANY\_MONTHLY**

**Table-valued function returning company names and count of demand for the given year and month**

**Input/Output**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE |  | DESCRIPTION / ATTRIBUTES |
| @ | RETURNS | TABLE TYPE |  | |
| @ | YEAR | INT | Input parameter for the table-valued function f\_GET\_DEMAND\_BY\_COMPANY\_MONTHLY. Enter a valid year. | |
| @ | MONTH | INT | Input parameter for the table-valued function f\_GET\_DEMAND\_BY\_COMPANY\_MONTHLY. Enter a valid month. | |

**Function: f\_GET\_DEMAND\_BY\_PRODUCT\_MONTHLY**

**Table-valued function returning product names and count of demand for the given year and month**

**Input/Output**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE |  | DESCRIPTION / ATTRIBUTES |
| @ | RETURNS | TABLE TYPE |  | |
| @ | YEAR | INT | Input parameter for the table-valued function f\_GET\_DEMAND\_BY\_PRODUCT\_MONTHLY. Enter a valid year. | |
| @ | MONTH | INT | Input parameter for the table-valued function f\_GET\_DEMAND\_BY\_PRODUCT\_MONTHLY. Enter a valid month. | |

**Function: f\_GET\_DEMAND\_BY\_PRODUCT\_ANNUAL**

**Table-valued function returning product names and count of demand for given years.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE |  | DESCRIPTION / ATTRIBUTES |
| @ | RETURNS | TABLE TYPE |  | |
| @ | Start\_Year | INT | Input parameter for the table-valued function f\_GET\_DEMAND\_BY\_PRODUCT\_ANNUAL. Enter a valid year. | |
| @ | End\_Year | INT | Input parameter for the table-valued function f\_GET\_DEMAND\_BY\_PRODUCT\_ANNUAL. Enter a valid year. | |

**Function: f\_GET\_DEMAND\_BY\_COMPANY\_ANNUAL**

**Table-valued function returning company names and count of demand for given years**

**Input/Output**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE |  | DESCRIPTION / ATTRIBUTES |
| @ | RETURNS | TABLE TYPE |  | |
| @ | Start\_Year | INT | Input parameter for the table-valued function f\_GET\_DEMAND\_BY\_COMPANY\_ANNUAL. Enter a valid year. | |
| @ | End\_Year | INT | Input parameter for the table-valued function f\_GET\_DEMAND\_BY\_COMPANY\_ANNUAL. Enter a valid year. | |

**Function: f\_COUNT\_OF\_COMPANY\_PRODUCT\_FOR\_COMPANYID**

**Scalar-valued function returning count of product for the given company id.**

**Input/Output**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE |  | DESCRIPTION / ATTRIBUTES |
| @ | RETURNS | TABLE TYPE |  | |
| @ | COMPANYID | INT | Input parameter for the scalar-valued function f\_COUNT\_OF\_COMPANY\_PRODUCT\_FOR\_COMPANYID. Enter a valid COMPANYID from the COMPANY table. | |

**Function: f\_FORMAT\_PHONE\_NUMBER**

**Scalar-valued function returning the true format for the given telephone number.**

**Input/Output**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE |  | DESCRIPTION / ATTRIBUTES |
| @ | RETURNS | TABLE TYPE |  | |
| @ | PHONENUMBER | VARCHAR(11) | Input parameter for the scalar-valued function [dbo].[Format\_Phone\_Number]. | |

**STORED PROCEDURES**

**Stored Procedure: sp\_CREATE\_COMPANY**

**Stored procedure using INSERT query to adding new company and address to company.**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | | DATA TYPE | DESCRIPTION |
| >@ | RoleId | int | Input parameter for the stored procedure sp\_CREATE\_COMPANY. Enter a RoleId int valid RoleId from the ROLE table. |
| >@ | Company\_Name | nvarchar(255) | Input parameter for the stored procedure sp\_CREATE\_COMPANY. Enter a Company\_Name nvarchar(255) |
| >@ | Website | nvarchar(100) | Input parameter for the stored procedure sp\_CREATE\_COMPANY. Enter a Website nvarchar(100) |
| >@ | Company\_Email | nvarchar(100) | Input parameter for the stored procedure sp\_CREATE\_COMPANY. Enter a Company\_Email nvarchar(100) |
| >@ | Fixed\_Number | nvarchar(15) | Input parameter for the stored procedure sp\_CREATE\_COMPANY. Enter a Fixed\_Number nvarchar(15) |
| >@ | Province\_Name | nvarchar(30) | Input parameter for the stored procedure sp\_CREATE\_COMPANY. Enter a Province\_Name nvarchar(30) valid Province\_Name from the PROVINCE table. |
| >@ | District\_Name | nvarchar(50) | Input parameter for the stored procedure sp\_CREATE\_COMPANY. Enter a District\_Name nvarchar(50) valid District\_Name from the DISTRICT table. |
| >@ | Address\_Line | nvarchar(255) | Input parameter for the stored procedure sp\_CREATE\_COMPANY. Enter a Address\_Line nvarchar(255) |

**Stored Procedure: sp\_ADD\_COMPANY\_ADDRESS**

**Stored procedure using INSERT query to adding new address to the company.**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | | DATA TYPE | DESCRIPTION |
| >@ | RoleId | int | Input parameter for the stored procedure sp\_ADD\_COMPANY\_ADDRESS. Enter a RoleId int valid RoleId from the ROLE table. |
| >@ | Company\_Name | nvarchar(255) | Input parameter for the stored procedure sp\_ADD\_COMPANY\_ADDRESS. Enter a Company\_Name nvarchar(255) valid Company\_Name from the COMPANY table. |
| >@ | Province\_Name | nvarchar(30) | Input parameter for the stored procedure sp\_ADD\_COMPANY\_ADDRESS. Enter a Province\_Name nvarchar(30) valid Province\_Name from the PROVINCE table. |
| >@ | District\_Name | nvarchar(50) | Input parameter for the stored procedure sp\_ADD\_COMPANY\_ADDRESS. Enter a District\_Name nvarchar(50) valid District\_Name from the DISTRICT table. |
| >@ | Address\_Line | nvarchar(255) | Input parameter for the stored procedure sp\_ADD\_COMPANY\_ADDRESS. Enter a Address\_Line nvarchar(255). |

**Stored Procedure:** **sp\_CREATE\_EMPLOYEE**

**Stored procedure using INSERT query to adding new employee and product to employee.**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | | DATA TYPE | DESCRIPTION |
| >@ | RoleForCreator | int | Input parameter for the stored procedure sp\_CREATE\_EMPLOYEE. Enter a RoleForCreator int valid ROLEID from the ROLE table. |
| >@ | Employee\_Name | nvarchar(50) | Input parameter for the stored procedure sp\_CREATE\_EMPLOYEE. Enter a Employee\_Name nvarchar(50) valid Employee\_Name from the EMPLOYEE table. |
| >@ | Employee\_Surname | nvarchar(50) | Input parameter for the stored procedure sp\_CREATE\_EMPLOYEE. Enter a Employee\_Surname nvarchar(50) valid Employee\_Surname from the EMPLOYEE table. |
| >@ | Mobile\_Number | nvarchar(15) | Input parameter for the stored procedure sp\_CREATE\_EMPLOYEE. Enter a Mobile\_Number nvarchar(15) valid Mobile\_Number from the EMPLOYEE table. |
| >@ | Employee\_Email | nvarchar(50) | Input parameter for the stored procedure sp\_CREATE\_EMPLOYEE. Enter a Employee\_Email nvarchar(50) valid Employee\_Email from the EMPLOYEE table. |
| >@ | ProfessionId | int | Input parameter for the stored procedure sp\_CREATE\_EMPLOYEE. Enter a ProfessionId int valid ProfessionId from the PROFESSION table. |
| >@ | DepartmentId | int | Input parameter for the stored procedure sp\_CREATE\_EMPLOYEE. Enter a DepartmentId int valid DepartmentId from the DEPARTMANT table. |
| >@ | RoleId | int | Input parameter for the stored procedure sp\_CREATE\_EMPLOYEE. Enter a RoleId int valid RoleId from the ROLE table. |
| >@ | ProductId | int | Input parameter for the stored procedure sp\_CREATE\_EMPLOYEE. Enter a ProductId int valid ProductId from the PRODUCT table. |

**Stored Procedure: sp\_CREATE\_PRODUCT**

**Stored procedure using INSERT query to adding new product.**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | | DATA TYPE | DESCRIPTION |
| >@ | RoleId | int | Input parameter for the stored procedure sp\_CREATE\_PRODUCT. Enter a RoleId int valid ROLEID from the ROLE table. |
| >@ | Product\_Name | nvarchar(100) | Input parameter for the stored procedure sp\_CREATE\_PRODUCT. Enter a Product\_Name nvarchar(100). |
| >@ | License\_PeriodId | int | Input parameter for the stored procedure sp\_CREATE\_PRODUCT. Enter a License\_PeriodId int valid License\_PeriodId from the LICENCE\_PERIOD table. |
| >@ | VersionNo | nvarchar(20) | Input parameter for the stored procedure sp\_CREATE\_PRODUCT. Enter a VersionNo nvarchar(20) valid ROLEID from the VERSION table. |

**Stored Procedure: sp\_ADD\_PRODUCT\_TO\_COMPANY**

**Stored procedure using INSERT query to adding new product to company.**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | | DATA TYPE | DESCRIPTION |
| >@ | RoleId | int | Input parameter for the stored procedure sp\_ADD\_PRODUCT\_TO\_COMPANY. Enter a RoleId int valid ROLEID from the ROLE table. |
| >@ | ProductId | int | Input parameter for the stored procedure sp\_ADD\_PRODUCT\_TO\_COMPANY. Enter a ProductId int valid ProductId from the PRODUCT table. |
| >@ | CompanyId | int | Input parameter for the stored procedure sp\_ADD\_PRODUCT\_TO\_COMPANY. Enter a CompanyId int valid CompanyId from the COMPANY table. |

**Stored Procedure: sp\_ASSIGN\_PRODUCT\_TO\_EMPLOYEE**

**Stored procedure using INSERT query to adding new product to employee.**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | | DATA TYPE | DESCRIPTION |
| >@ | RoleId | int | Input parameter for the stored procedure sp\_ASSING\_PRODUCT\_TO\_EMPLOYEE. Enter a RoleId int valid ROLEID from the ROLE table. |
| >@ | ProductId | int | Input parameter for the stored procedure sp\_ASSING\_PRODUCT\_TO\_EMPLOYEE. Enter a ProductId int valid ProductId from the PRODUCT table. |
| >@ | EmployeeId | int | Input parameter for the stored procedure sp\_ASSING\_PRODUCT\_TO\_EMPLOYEE. Enter a EmployeeId int valid EmployeeId from the EMPLOYEE table. |

**Stored Procedure: sp\_ASSIGN\_PRODUCT\_TO\_EMPLOYEE\_MULTIPLE**

**Stored procedure using INSERT query to adding new products to the company.**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | | DATA TYPE | DESCRIPTION |
| >@ | RoleId | int | Input parameter for the stored procedure sp\_ASSIGN\_PRODUCT\_TO\_EMPLOYEE\_MULTIPLE. Enter a RoleId int valid RoleId from the ROLE table. |
| >@ | Products | nvarchar(max) | Input parameter for the stored procedure sp\_ASSIGN\_PRODUCT\_TO\_EMPLOYEE\_MULTIPLE. Enter a Products nvarchar(max). |
| >@ | EmployeeId | int | Input parameter for the stored procedure sp\_ASSIGN\_PRODUCT\_TO\_EMPLOYEE\_MULTIPLE. Enter a EmployeeId int valid EmployeeId from the EMPLOYEE table. |

**Stored Procedure: sp\_CREATE\_PROFFESION**

**Stored procedure using INSERT query to add profession.**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | | DATA TYPE | DESCRIPTION |
| >@ | RoleId | int | Input parameter for the stored procedure sp\_CREATE\_PROFESSION. Enter a RoleId int valid RoleId from the ROLE table. |
| >@ | Profession\_Name | nvarchar(50) | Input parameter for the stored procedure sp\_CREATE\_PROFESSION. Enter a Profession\_Name nvarchar(50) valid Profession\_Name from the ROLE table. |

**Stored Procedure: sp\_CREATE DEPARTMENT**

**Stored procedure using INSERT query to add department.**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | | DATA TYPE | DESCRIPTION |
| >@ | RoleId | int | Input parameter for the stored procedure sp\_CREATE\_DEPARTMENT. Enter a RoleId int valid RoleId from the ROLE table. |
| >@ | Department\_Name | nvarchar(50) | Input parameter for the stored procedure sp\_CREATE\_DEPARTMENT. Enter a Department\_Name nvarchar(50) valid Department\_Name from the DEPARTMENT table. |

**Stored Procedure: sp\_ASSIGN\_DEMAND**

**Stored procedure using UPDATE query to assign demand to employee.**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | | DATA TYPE | DESCRIPTION |
| >@ | RoleId | int | Input parameter for the stored procedure sp\_ASSIGN\_DEMAND. Enter a RoleId int valid RoleId from the ROLE table. |
| >@ | ProductId | int | Input parameter for the stored procedure sp\_ASSIGN\_DEMAND. Enter a ProductId int valid ProductId from the PRODUCT table. |
| >@ | EmployeeId | int | Input parameter for the stored procedure sp\_ASSIGN\_DEMAND. Enter a EmployeeId int valid EmployeeId from the EMPLOYEE table. |

**Stored Procedure: sp\_CLOSE\_DEMAND**

**Stored procedure using UPDATE query to closing demand.**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | | DATA TYPE | DESCRIPTION |
| >@ | DemandId | int | Input parameter for the stored procedure sp\_CLOSING\_DEMAND. Enter a DemandId int valid DemandId from the DEMAND table. |
| >@ | EmployeeId | int | Input parameter for the stored procedure sp\_CLOSING\_DEMAND. Enter a EmployeeId int valid EmployeeId from the EMPLOYEE table. |
| >@ | RoleId | int | Input parameter for the stored procedure sp\_CLOSING\_DEMAND. Enter a RoleId int valid RoleId from the ROLE table. |
| >@ | Closing\_statement | nvarchar(max) | Input parameter for the stored procedure sp\_CLOSING\_DEMAND. Enter a Closing\_statement nvarchar(max). |
| >@ | Solved\_hour | int | Input parameter for the stored procedure sp\_CLOSING\_DEMAND. Enter a Solved\_hour int. |

**Stored Procedure: sp\_CREATE\_VERSION**

**Stored procedure using INSERT query for adding new version.**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | | DATA TYPE | DESCRIPTION |
| >@ | RoleId | int | Input parameter for the stored procedure sp\_CREATE\_VERSION. Enter a RoleId int valid RoleId from the ROLE table. |
| >@ | ProductId | int | Input parameter for the stored procedure sp\_CREATE\_VERSION. Enter a ProductId int valid ProductId from the PRODUCT table. |
| >@ | VersionNo | nvarchar(20) | Input parameter for the stored procedure sp\_CREATE\_VERSION. Enter a VersionNo nvarchar(20) valid VersionNo from the VERSION table. |

**Stored Procedure: sp\_CREATE\_COMPANY\_USER**

**Stored procedure using INSERT query to add new company user.**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | | DATA TYPE | DESCRIPTION |
| >@ | RoleIdForCreator | int | Input parameter for the stored procedure sp\_CREATE\_COMPANY\_USER. Enter a RoleIdForCreator int valid RoleIdForCreator from the ROLE table. |
| >@ | Company\_User\_Name | nvarchar(50) | Input parameter for the stored procedure sp\_CREATE\_COMPANY\_USER. Enter a Company\_User\_Name nvarchar(50). |
| >@ | Company\_User\_Surname | nvarchar(50) | Input parameter for the stored procedure sp\_CREATE\_COMPANY\_USER. Enter a Company\_User\_Surname nvarchar(50). |
| >@ | Mobile\_Number | nvarchar(15) | Input parameter for the stored procedure sp\_CREATE\_COMPANY\_USER. Enter a Mobile\_Number nvarchar(15). |
| >@ | Company\_User\_Email | nvarchar(50) | Input parameter for the stored procedure sp\_CREATE\_COMPANY\_USER. Enter a Company\_User\_Email nvarchar(50). |
| >@ | CompanyId | int | Input parameter for the stored procedure sp\_CREATE\_COMPANY\_USER. Enter a CompanyId int valid CompanyId from the COMPANY table. |
| >@ | RoleId | int | Input parameter for the stored procedure sp\_CREATE\_COMPANY\_USER. Enter a RoleId int valid RoleId from the ROLE table. |

**Stored Procedure: sp\_CREATE\_DEMAND**

**Stored procedure using INSERT query to adding new demand.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | | DATA TYPE | DESCRIPTION | |
| >@ | RoleId | int | Input parameter for the stored procedure sp\_CREATE\_DEMAND. Enter a RoleId int valid RoleId from the ROLE table. | |
| >@ | Title | nvarchar(max) | Input parameter for the stored procedure sp\_CREATE\_DEMAND. Enter a Title nvarchar(max). | |
| >@ | Text | nvarchar(max) | Input parameter for the stored procedure sp\_CREATE\_DEMAND. Enter a Text nvarchar(max). | |
| >@ | Demand\_TypeId | int | Input parameter for the stored procedure sp\_CREATE\_DEMAND. Enter a Demand\_TypeId int valid Demand\_TypeId from the DEMAND\_TYPE table. | |
| >@ | Order\_Of\_UrgencyId | int | Input parameter for the stored procedure sp\_CREATE\_DEMAND. Enter a Order\_Of\_UrgencyId int valid Order\_Of\_UrgencyId from the ORDER\_OF\_URGENCY table. | |
| >@ | Company\_UserId | int | Input parameter for the stored procedure sp\_CREATE\_DEMAND. Enter a Company\_UserId int valid Company\_UserId from the COMPANY\_USER table. |
| >@ | ProductId | int | Input parameter for the stored procedure sp\_CREATE\_DEMAND. Enter a ProductId int valid ProductId from the PRODUCT table. |
| >@ | VersionId | int | Input parameter for the stored procedure sp\_CREATE\_DEMAND. Enter a VersionId int valid VersionId from the VERSION table. |

**Stored Procedure: sp\_UPDATE\_DEMAND**

**Stored procedure using UPDATE query to update demand.**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | | DATA TYPE | DESCRIPTION |
| >@ | RoleId | int | Input parameter for the stored procedure sp\_UPDATE\_DEMAND. Enter a RoleId int valid RoleId from the ROLE table. |
| >@ | DemandId | int | Input parameter for the stored procedure sp\_UPDATE\_DEMAND. Enter a DemandId int valid DemandId from the DEMAND table. |
| >@ | Title | nvarchar(max) | Input parameter for the stored procedure sp\_UPDATE\_DEMAND. Enter a Title nvarchar(max). |
| >@ | NewText | nvarchar(max) | Input parameter for the stored procedure sp\_UPDATE\_DEMAND. Enter a NewText nvarchar(max). |
| >@ | Demand\_TypeId | int | Input parameter for the stored procedure sp\_UPDATE\_DEMAND. Enter a Demand\_TypeId int valid Demand\_TypeId from the DEMAND\_TYPE table. |
| >@ | Order\_Of\_UrgencyId | int | Input parameter for the stored procedure sp\_UPDATE\_DEMAND. Enter a Order\_Of\_UrgencyId int valid Order\_Of\_UrgencyId from the ORDER\_OF\_URGENCY table. |
| >@ | Product\_Name | nvarchar(100) | Input parameter for the stored procedure sp\_UPDATE\_DEMAND. Enter a Product\_Name nvarchar(100) valid Product\_Name from the PRODUCT table. |
| >@ | VersionNo | nvarchar(20) | Input parameter for the stored procedure sp\_UPDATE\_DEMAND. Enter a VersionNo nvarchar(20) valid VersionNo from the VERSION table. |
| >@ | Record\_Status | bit | Input parameter for the stored procedure sp\_UPDATE\_DEMAND. Enter a Record\_Status bit. |

**VIEWS**

**View: vw\_GET\_ALL\_COMPANY\_USER**

**Company User names and surnames.**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | DATA TYPE | N | DESCRIPTION |
| NAME | nvarchar(50) |  | Company User name FROM COMPANY\_USER |
| SURNAME | nvarchar(50) |  | Company User surname FROM COMPANY\_USER |

**View: vw\_GET\_ALL\_COMPANY**

**Company name, email, website and phone number**.

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | DATA TYPE | N | DESCRIPTION |
| ID | int |  | Company ID FROM COMPANY |
| COMPANY\_NAME | nvarchar(255) |  | Company Name FROM COMPANY |
| WEBSITE | nvarchar(100) |  | Company Website FROM COMPANY |
| EMAIL | nvarchar(100) |  | Company Email FROM COMPANY |
| FIXED\_NUMBER | nvarchar(15) | **N** | Company Fixed Number FROM COMPANY |

**View: vw\_GET\_ALL\_DEMAND**

**All Demand columns.**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | DATA TYPE | N | DESCRIPTION |
| ID | int |  | Demand ID FROM DEMAND |
| TITLE | nvarchar(max) |  | Demand title FROM DEMAND |
| TEXT | nvarchar(max) |  | Demand text FROM DEMAND |
| DEMAND\_TYPEID | int |  | Demand type id FROM DEMAND |
| ORDER\_OF\_URGENCYID | int |  | Demand order of urgency FROM DEMAND |
| NAME | DATA TYPE | N | DESCRIPTION |
| DEMAND\_STATEID | int | **N** | Demand statement id FROM DEMAND |
| CLOSING\_STATEMENT | nvarchar(max) | **N** | Demand closing statement FROM DEMAND |
| CLOSING\_DATE | datetime | **N** | Demand closing date FROM DEMAND |
| COMPANY\_USERID | int |  | Demand company user id FROM DEMAND |
| EMPLOYEEID | int | **N** | Demand employee id FROM DEMAND |
| PRODUCTID | int |  | Demand product id FROM DEMAND |
| SOLVEDHOUR | int | **N** | Demand solved hour FROM DEMAND |
| VERSIONID | int |  | Demand version id FROM DEMAND |
| CREATE\_DATE | datetime |  | Demand create date FROM DEMAND |
| MODIFICATION\_DATE | datetime |  | Demand modification date FROM DEMAND |
| RECORD\_STATUS | bit |  | Demand record status FROM DEMAND |

**View: vw\_GET\_ALL\_PRODUCT**

**Return product id, name, license period and version with JOIN statements.**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | DATA TYPE | N | DESCRIPTION |
| ID | int |  | Product id from PRODUCT |
| PRODUCT\_NAME | nvarchar(100) |  | Product name from PRODUCT |
| PERIOD | nvarchar(50) |  | Product license period from LICENSE\_PERIOD |
| VERSIONNO | nvarchar(20) |  | Product version no from VERSION |