Governors State University

CPSC 8845 – 03

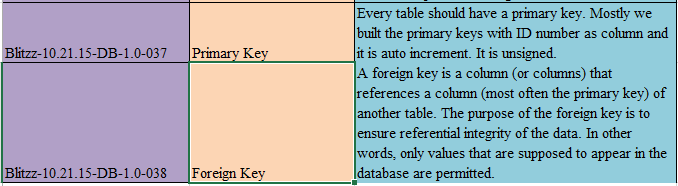
Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Rev | Date | Purpose | Originator |
| Draft | 12-2-2015 | Final Draft | Ujjwal Kumar Chowdary Konka (UJ) |
|  |  |  | Hima Bindu Jampani (Bindu) |
|  |  |  | Nagaraju Deshini (Nag) |
|  |  |  | Mohammed Abdul Lateef (Lateef) |

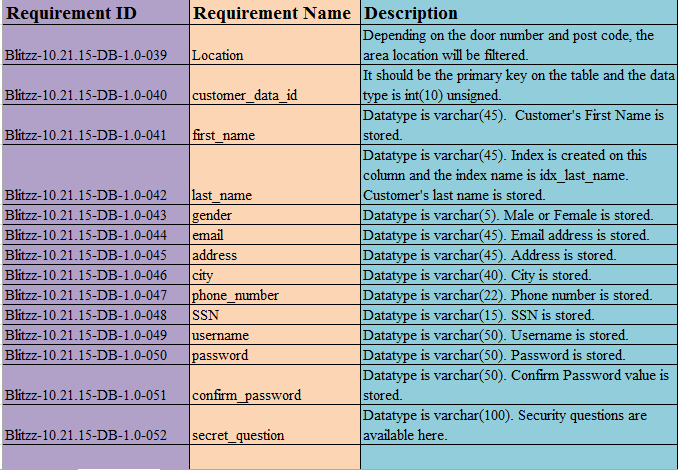
# **Feature Requirements**

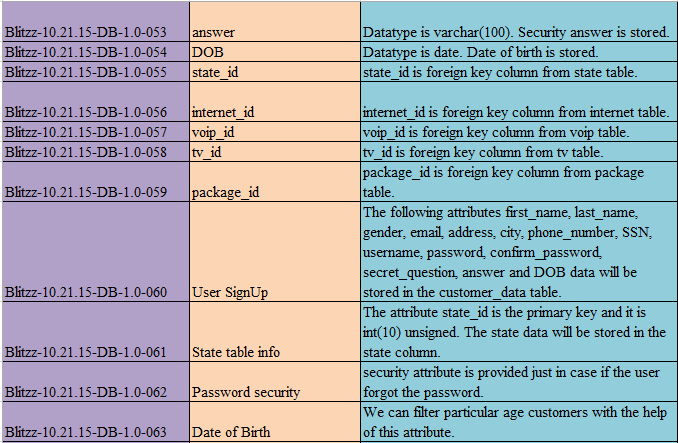
1. **General Database Requirements.**

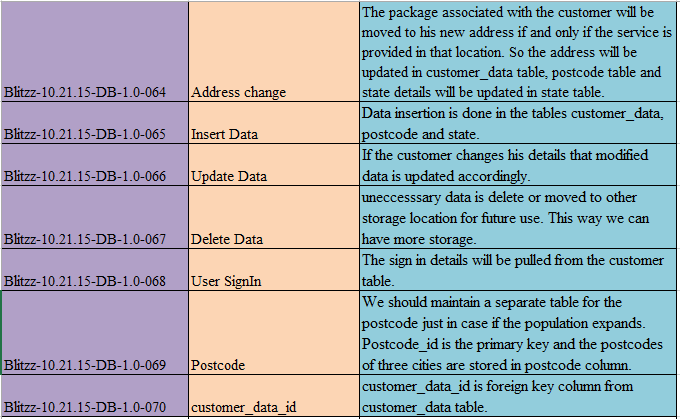
|  |  |  |
| --- | --- | --- |
| **Requirement ID** | **Requirement Name** | **Description** |
| Blitzz-10.21.15-DB-1.0-001 | Create Database | You must have **create data base** permission on the data base group that will contain the data base and **use template** permission on at least one data base configuration **template** |
| Blitzz-10.21.15-DB-1.0-002 | Create snapshort | Take data base snap short from time to time |
| Blitzz-10.21.15-DB-1.0-003 | Daily DB backups and maintainance | Take regular backups and the backup should be performed with in 12 pm to 2 am CST. Store the backups in separate destination where your database is not stored. |
| Blitzz-10.21.15-DB-1.0-004 | Database Integrity | A weekly routine maintenance should be run to fix and correct any Database issues. |
| Blitzz-10.21.15-DB-1.0-005 | Duplex System | Our company will maintain a mirror system and this will be switched in less than 30 sec so that all the data will be updated in real time and make sure both the systems are in sync all the time. |
| Blitzz-10.21.15-DB-1.0-006 | Fail Over case | In order to handle this case the system must support at least 30 sec worth of data in temporary memory space (5000 customer records). |
| Blitzz-10.21.15-DB-1.0-007 | Delete snapshot and manage retention time | Delete unnecessary snapshots |
| Blitzz-10.21.15-DB-1.0-008 | Recovery | Recover the data base from the backups or snapshots. Perform DB recovery n case of failure, it is better to use RMAN for online recovery where your instances is still in up and running mode. |
| Blitzz-10.21.15-DB-1.0-009 | Database storage allocation | We will estimate how much data will be stored in the database. Consider the row length of each table, number of business transaction that the database will need to store over a given period of time. Plan for the amount of space required for the indexes as they also occupy or even more than that of the table. Keep in mind the RAID configuration. |
| Blitzz-10.21.15-DB-1.0-010 | Disk Speed | While a disk system may be of sufficient size to cope with predicted requirements, it may not be able to handle the anticipated number of IO's per second, generated by the work load. So here we may need some extra disk. |
| Blitzz-10.21.15-DB-1.0-011 | Disk Utilization | Sizing Temp DB at the transaction log- both can vary in size quite a wide range. |
| Blitzz-10.21.15-DB-1.0-012 | Point in time recovery storage allocation | Consider the number of users and average expected number of transactions in a particular time period and include room for growth. Also calculate additional storage to accommodate the point in time recovery right-ahead logs. |
| Blitzz-10.21.15-DB-1.0-013 | Database group | Create the database groups as they provide CPU, memory, Storage and network resources required to run the database. |
| Blitzz-10.21.15-DB-1.0-014 | DB Performance | The DB performance should not degrade less than 99.99% |
| Blitzz-10.21.15-DB-1.0-015 | Peoplesoft | This will be our HR database |
| Blitzz-10.21.15-DB-1.0-016 | Retrieving Data | The data should be easy to retrieve data and query performance should be high |
| Blitzz-10.21.15-DB-1.0-017 | Adding Data | The data should be easy to add data and query performance should be high |
| Blitzz-10.21.15-DB-1.0-018 | Deleting Data | The data should be easy to delete data and query performance should be high |
| Blitzz-10.21.15-DB-1.0-019 | Updating Data | The data should be easy to update data and query performance should be high |
| Blitzz-10.21.15-DB-1.0-020 | Index | Create indexes for fast query performance. |
| Blitzz-10.21.15-DB-1.0-021 | Integriy constraints | Create integrity constraints ( NOT NULL, Unique, Primary, Foreign, Check) for establishing RDBMS. |
| Blitzz-10.21.15-DB-1.0-022 | Views | Create views for security purpose. |
| Blitzz-10.21.15-DB-1.0-023 | Triggers | Create triggers for auto updating. |
| Blitzz-10.21.15-DB-1.0-024 | Data files | Create enough data files at physical level which will have table spaces, segments, extends and data blocks at logical level. |
| Blitzz-10.21.15-DB-1.0-025 | Partitions | Create partitions for better performance. |
| Blitzz-10.21.15-DB-1.0-026 | Materialized View | It is better to create materialized views as we can quickly see the changes done at server side without firing any lengthy commands or query. It has got 3 refresh modes, complete, fast and force. |
| Blitzz-10.21.15-DB-1.0-027 | Monitoring of Database | Always check the alert log files. |
| Blitzz-10.21.15-DB-1.0-028 | User management | Creating users and manage users. Like grant privileges revoke privilege, lock user account, unlock user account, change user password, restrict resources. |
| Blitzz-10.21.15-DB-1.0-029 | Upgradation | Plan for the upgradation of Database in future. |
| Blitzz-10.21.15-DB-1.0-030 | Migration | Plan for migration of the DB from one operating to another OS. |
| Blitzz-10.21.15-DB-1.0-031 | Cloning | clone the database. Make some copy for testing purpose. |
| Blitzz-10.21.15-DB-1.0-032 | Patches | Download and install latest patches and fix the bugs. |
| Blitzz-10.21.15-DB-1.0-033 | Tuning | Tune database performance. For example, check with the degradation of performance. |
| Blitzz-10.21.15-DB-1.0-034 | Database Availability | Database should be available 24/7. There shouldn't be any down time for the users and the database. |
| Blitzz-10.21.15-DB-1.0-035 | Mutliplexing of control file | You should make a copy of the control file. |
| Blitzz-10.21.15-DB-1.0-036 | Mutliplexing of redo log files | Redo log file is physical file which is used to store latest transaction information. So it is always better to maintain copis of Redo log files. |

****

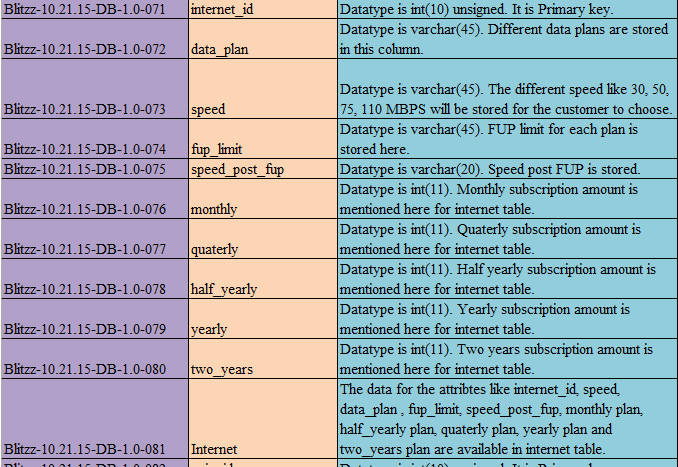
1. **Customer Database Requirements.**

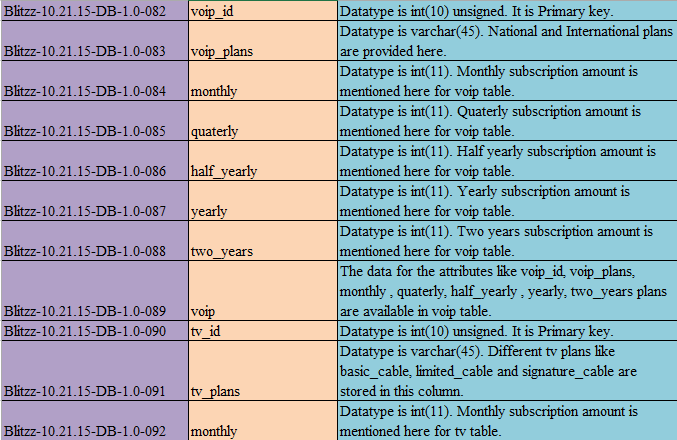
****

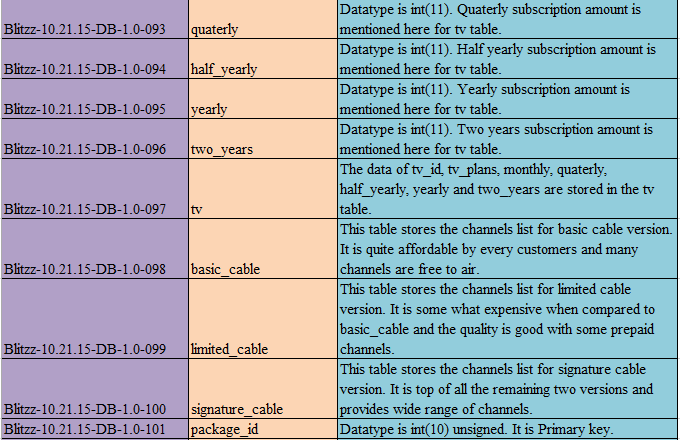
****

****

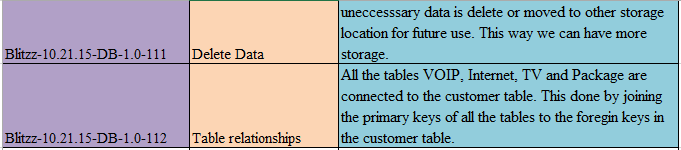
1. **Service Database Requirements.**

****

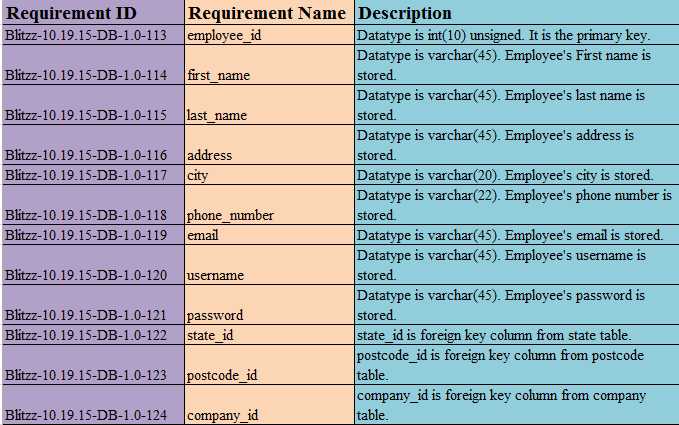
****

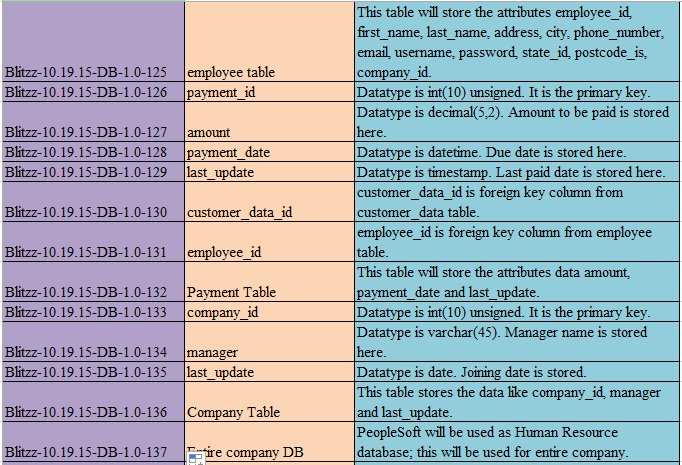
****

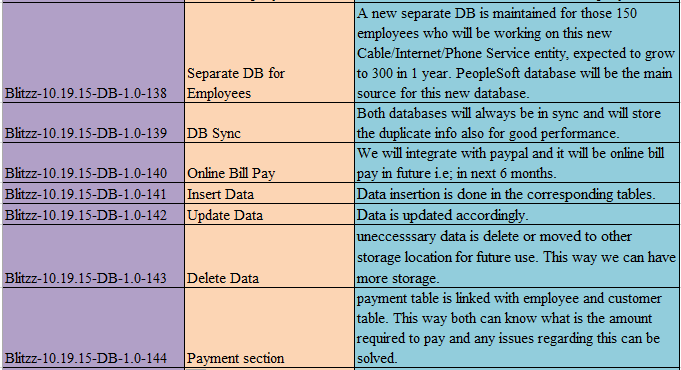
****

****

1. **Employee Database Requirements.**

****

****

****

# **Design Description**

The Design that we are using in our project will have three databases to maintain the website. Those three databases are Employee Database, Customer Database and Services Database. We need to have all three of the databases for successful functionality. For example, let us assume a scenario: when a customer registers for a package of VOIP, Internet and TV then the operations that happens in the database are:

1. He selects the packages from the services database.
2. He enters his info for the registration which is stored in customer database.
3. He pays for the packages which is stored in the payment section of employee database. In future if there is a complaint then all the info about customer and his package is gathered by Employee to solve the problem.

Here the data flows from Services database to customer database and then finally into employee database.

VOIP

Internet

TV cable

Data is pulled from Services Database

Package Selection

Customer

Address

City

Stored in Customer database

Customer details

Employee

Emp\_address

Payment

Company

Stored in Employee database

Employee

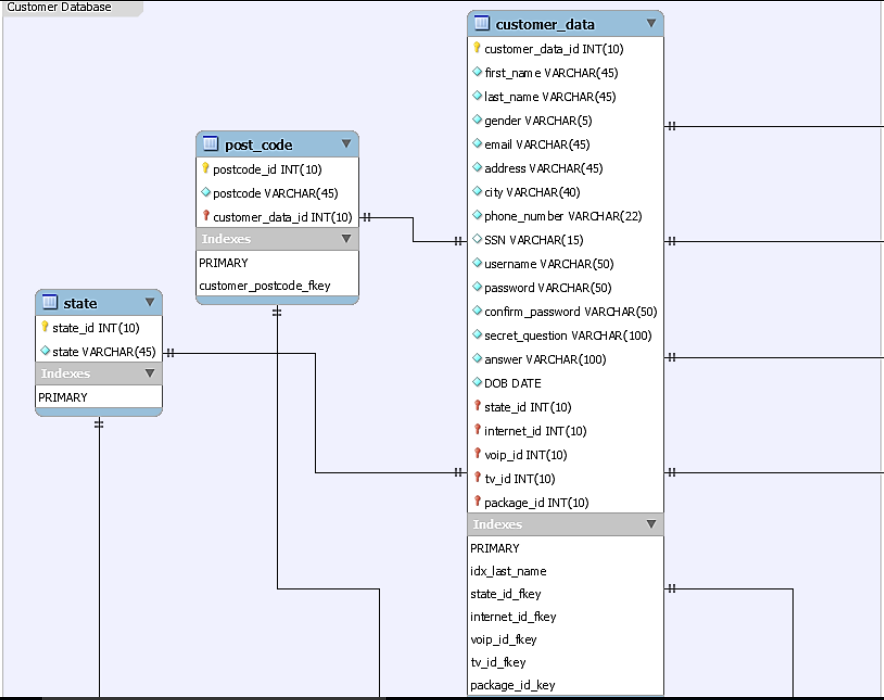
# 

# **Internal/external Interface Impacts and Specification**

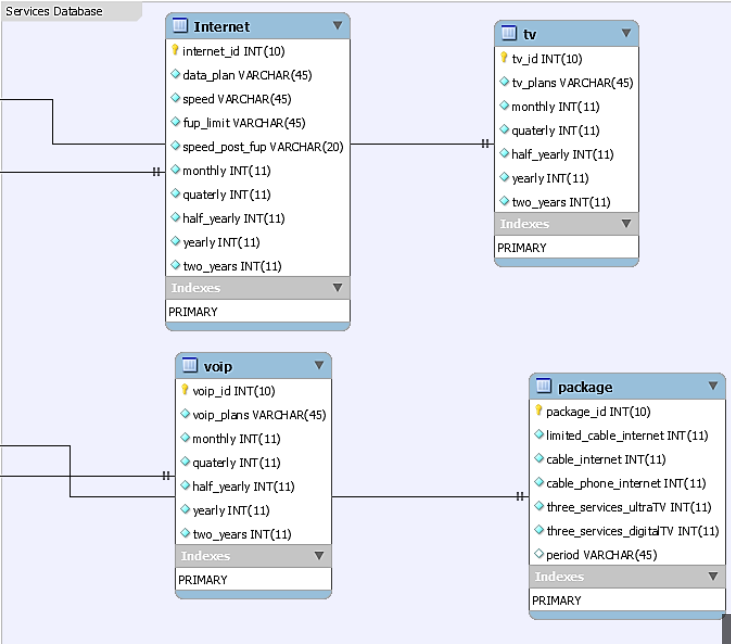
The more changes that we will be making in are at Package table. We have to develop a snowflake schema for the packages table in the services database to be persistent in detail. This is very important for Internal and external interface impacts as this will have the wide range of package information that will be displayed on the website.

# **Design Units Impacts**

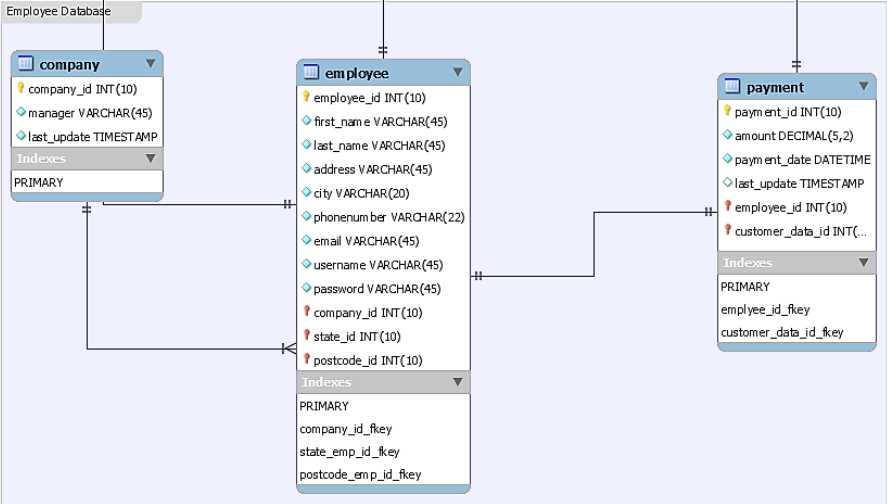
**ER Model of Customer Database:**

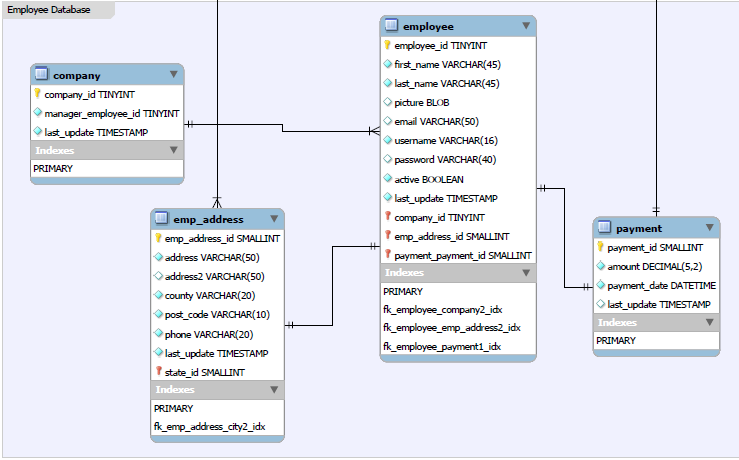
****

**ER Model of Services Database:**

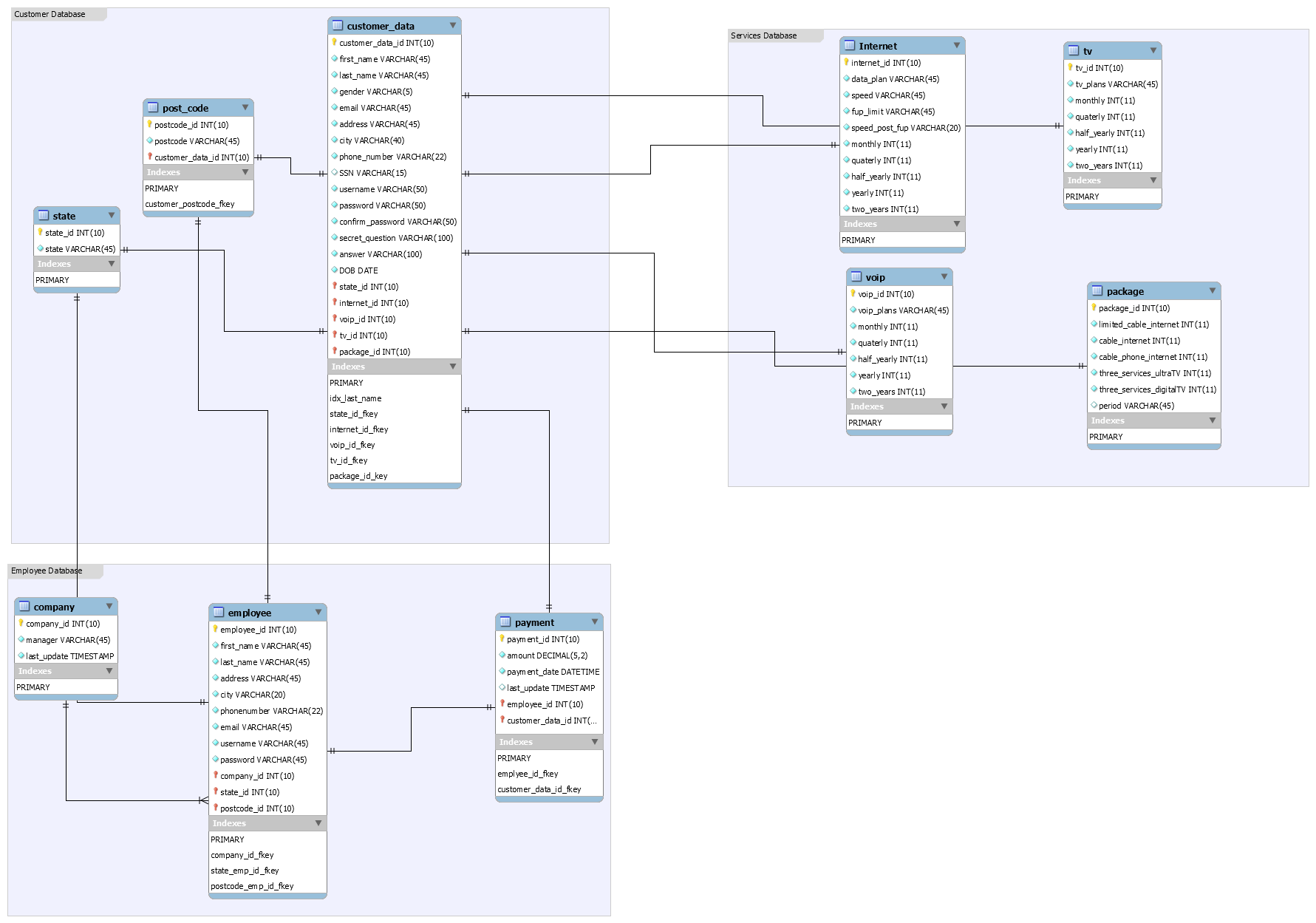
****

**ER Model of Employee Database:**

****



**EER Model of the Project:**

****