# **Physical Schema**

A physical schema is a representation of a data design as implemented, or intended to be implemented, in a database management system. Before implementing the real database by using MySQL RDBMS, we will implement the specifications for each table. Specifications includes the name, variable which is the abbreviation of the name, data type, size, whether the key is primary key or foreign key, and whether the key is the index or not. In this paper, we also discuss about the integrity of each key, mainly about entity and referential integrity.

# **Camping Car**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) |
| 1 | Camping Car ID | CIN | VARCHAR | 17 | **☑** |  |
| 2 | Issue date | ISSUE\_DT | DATE | 3 | **☑** |  |
| 3 | Car type | CAR\_TP | VARCHAR | 25 |  |  |
| 4 | Manufacture date | MANU\_DT | DATE | 3 |  |  |
| 5 | Distance Driven | DRI\_DIS | INTEGER | 4 |  |  |
| 6 | Facilities | CAMP\_FAC | VARCHAR | 50 |  |  |
| 7 | Passenger | CAMP\_CAP | INTEGER | 4 |  |  |
| 8 | Availability | CAMP\_FL | TINYINT | 1 |  |  |
| 9 | Required License | LCNS\_RQ | VARCHAR | 10 |  |  |
| 10 | Branch ID | BIN | VARCHAR | 18 |  | **☑** |

Table 1 - CAMP\_CAR

Table 1 shows the camping car table. This table exists to manage the information about camping cars. It has the set of CIN (Camping Car ID) and ISSUE\_DT (Issue Date) as a primary key. We assigned this set of the key as a primary key because of the regulation of car number pad assignment. Purpose of this regulation is to avoid the misuse of number pad in driving stolen car. Therefore, to identify the camping car, or just a car, we need to search it with car number and its issued date. The BIN (Branch ID) key in this table is the foreign key that references the Branch ID from the branch table. In the aspect of entity integrity, the set of CIN (Camping Car ID) and ISSUE\_DT (Issue Date) should be unique and not null. In the aspect of referential integrity, BIN (Branch ID), which references Branch ID from branch table, should be casually updated if the referenced key is updated. Also, if the referenced key is deleted, the we denote that the camping car is turned in to the head branch. Therefore, we will set the BIN (Branch ID) into head branch if the referenced key is deleted.

# **Customer**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) |
| 1 | License number | LCNS\_NO | VARCHAR | 12 | **☑** |  |
| 2 | First name | FNAME | VARCHAR | 50 |  |  |
| 3 | Middle name | MNAME | VARCHAR | 50 |  |  |
| 4 | Last name | LNAME | VARCHAR | 50 |  |  |
| 5 | Phone | CUS\_PHN | VARCHAR | 11 |  |  |
| 6 | Email | CUS\_EML | VARCHAR | 50 |  |  |
| 7 | Address | CUS\_ADDR | VARCHAR | 50 |  |  |
| 8 | Age | CUS\_AGE | INTEGER | 4 |  |  |
| 9 | Login ID | LOGIN\_ID | VARCHAR | 13 |  | **☑** |

Table 2 - CUSTOMER

Table 2 shows the customer table. This table stores the information about customers. It has the LCNS\_NO (License Number) as a primary key, and LOGIN\_ID (Login ID) as a foreign key that references the Login ID in customer credential table. In the aspect of the entity integrity, LCNS\_NO (License Number) should be unique and not null. In the aspect of referential integrity, LOGIN\_ID (Login ID) must be causally updated and deleted if the reference key is updated and deleted.

# **Customer Credential**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) |
| 1 | Login ID | LOGIN\_ID | VARCHAR | 13 | **☑** |  |
| 2 | Password | PASSWORD | VARCHAR | 13 |  |  |
| 3 | Login time | LOGIN\_TIME | TIME | 3 |  |  |
| 4 | Logout time | LOGOUT\_TIME | TIME | 3 |  |  |

Table 3 - CUSTOMER\_CREDENTIAL

Table 3 shows the customer credential table. Customer credential table stores the information of the logging data of customers, which includes ID, Password, and Login and Logout time. It has LOGIN\_ID (Login ID) as a primary key.

# **Event**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) |
| 1 | Event ID | EIN | INTEGER AUTO INCREMENT | 4 | **☑** |  |
| 2 | Start date | EVNT\_SDT | DATE | 3 |  |  |
| 3 | End date | EVNT\_EDT | DATE | 3 |  |  |
| 4 | Description | EVNT\_DES | VARCHAR | 100 |  |  |
| 5 | Applied Location | APP\_LOC | VARCHAR | 50 |  |  |

Table 4 - EVENT

Table 4 shows the event table, which contains the information of event that is held from particular start date to end date. This table has EIN (Event ID) as a primary key, and it is an auto incremented number from 1 to the end of the tuple.

# **Accident**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) |
| 1 | Accident ID | AIN | VARCHAR | 10 | **☑** |  |
| 2 | Camping Car ID | CIN | VARCHAR | 17 |  | **☑** |
| 3 | Issue date | ISSUE\_DT | DATE | 3 |  | **☑** |
| 4 | Accident date | ACC\_DT | DATE | 3 |  |  |
| 5 | Description | ACC\_DES | VARCHAR | 50 |  |  |
| 6 | Damage | DAMAGE | FLOAT | 4 |  |  |

Table 5 - ACCIDENT

Table 5 shows the accident table. This table includes the information of camping car accident. It has the set of AIN (Accident ID), CIN (Camping Car ID), and ISSUE\_DT (Issue Date) as a primary key. Due to the entity integrity, AIN (Accident ID) should be unique and not null. In the aspect of referential integrity, foreign keys, which are CIN (Camping Car ID) and ISSUE\_DT (Issue Date), should be casually updated and deleted when the following referenced keys are updated and deleted because the accident data is dependent to the camping car, which also means that accident information cannot be exists without the camping car.

# **Review**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) |
| 1 | Review ID | REV\_SQ | INTEGER AUTO INCREMENT | 4 | **☑** |  |
| 2 | Rate of the service | RATE | INTEGER | 4 |  |  |
| 3 | Description | REV\_DES | VARCHAR | 100 |  |  |
| 4 | License number | LCNS\_NO | VARCHAR | 12 |  | **☑** |
| 5 | Camping Car ID | CIN | VARCHAR | 17 |  | **☑** |
| 6 | Issue date | ISSUE\_DT | DATE | 3 |  | **☑** |

Table 6 - REVIEW

Table 6 shows the review table that stores the data of reviews for each camping car. It has REV\_SQ (Review Number), which is auto incremented number, as a primary key. Following table generates the relation between customer and camping car table. Therefore, it contains the LCNS\_NO (License Number), CIN (Camping Car ID), ISSUE\_DT (Issue Date) as foreign keys that each reference the License Number from customer table, and Camping Car ID and ISSUE\_DT from camping car table. In the aspect of entity integrity, REV\_SQ (Review ID) should be unique and not null. LCNS\_NO (License Number) should be converted into the default value of “Unknown”, if the referenced key is deleted, because the review should be remained even though customer information is deleted. If the following key is updated. LCNS\_NO (License Number) of the review table is updated casually. The set of CIN (Camping Car ID) and ISSUE\_DT (Issue Date) is updated and deleted casually if the referenced keys are updated and deleted.

# **Accessory**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) |
| 1 | Accessory ID | ACSRY\_NO | VARCHAR | 10 | **☑** |  |
| 2 | Status | STATUS | TINYINT | 1 |  |  |
| 3 | Amount | ACSRY\_AMT | INTEGER | 4 |  |  |
| 4 | Type | ACSRY\_TP | VARCHAR | 50 |  |  |

Table 7 - ACCESSORY

Table 7 shows the accessory table, which includes the information of accessories. It has primary key of ACSRY\_NO (Accessory ID) which follows the entity integrity. Therefore, it should be unique and not null.

# **Reservation**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) |
| 1 | Reservation ID | RID | VARCHAR | 20 | **☑** |  |
| 2 | Start date | RES\_SDT | DATE | 3 |  |  |
| 3 | End date | RES\_EDT | DATE | 3 |  |  |
| 4 | Start location | RES\_SLOC | VARCHAR | 50 |  |  |
| 5 | End location | RES\_ELOC | VARCHAR | 50 |  |  |
| 6 | Passenger | PASS\_AMT | INTEGER | 4 |  |  |
| 7 | Meter start | METR\_STD | INTEGER | 4 |  |  |
| 8 | Meter end | METR\_END | INTEGER | 4 |  |  |
| 9 | Rental amount | RENT\_AMT | FLOAT | 4 |  |  |
| 10 | Additional amount | ADD\_AMT | FLOAT | 4 |  |  |
| 11 | Total amount | TTL\_AMT | FLOAT | 4 |  |  |
| 12 | Penalty amount | PNLT\_AMT | FLOAT | 4 |  |  |
| 13 | Billing address | BILL\_ADDR | VARCHAR | 50 |  |  |
| 14 | Payment ID | PAY\_ID | VARCHAR | 10 |  | **☑** |
| 15 | License Number | LCNS\_NO | VARCHAR | 12 |  | **☑** |
| 16 | Camping Car ID | CIN | VARCHAR | 17 |  | **☑** |
| 17 | Issue date | ISSUE\_DT | DATE | 3 |  | **☑** |

Table 8 - RESERVATION

Table 8 shows the reservation table. It contains the information of reservations of the camping car. It has RID (Reservation ID) as a primary key. PAY\_ID (Payment ID), LCNS\_NO (License Number), CIN (Camping Car ID), and ISSUE\_DT (Issue Date) are the foreign keys that reference each Payment ID from pay table, License Number from customer, and Camping Car ID and Issue Date from the camping car table. In the aspect of entity integrity, RID (Reservation ID) should be unique and not null. In the aspect of referential integrity, PAY\_ID (Payment ID) should be changed into the “refund” if referenced key is deleted, because deletion on the payment table means the refund of the reservation. If referenced key is update, following foreign key should be updated casually. LCNS\_NO (License Number), CIN (Camping Car ID), and ISSUE\_DT (Issue Date) should be updated and deleted casually when each referenced key is updated and deleted.

# **Branch**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) |
| 1 | Branch ID | BIN | VARCHAR | 18 | **☑** |  |
| 2 | State | STATE | VARCHAR | 20 |  |  |
| 3 | Street | STREET | VARCHAR | 20 |  |  |
| 4 | Zip | ZIP | VARCHAR | 5 |  |  |
| 5 | Email | BRN\_EML | VARCHAR | 50 |  |  |
| 6 | Phone | BRN\_PHN | VARCHAR | 11 |  |  |
| 7 | Capacity | BRN\_CAP | INTEGER | 4 |  |  |

Table 9 - Branch

Table 9 shows the branch table, which stores the information of branch at each location that owns the camping car. This table as BIN (Branch ID) as a primary key, which fulfills the entity integrity. In short, following key is unique and not null.

# **Insurance**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) |
| 1 | Insurance ID | IIN | VARCHAR | 10 | **☑** |  |
| 2 | Type | INS\_TP | VARCHAR | 15 |  |  |
| 3 | Collision coverage | CLSN\_COV | TINYINT | 1 |  |  |
| 4 | Body coverage | BODY\_COV | TINYINT | 1 |  |  |
| 5 | Medical coverage | MEDI\_COV | TINYINT | 1 |  |  |
| 6 | Price | INS\_PRC | FLOAT | 4 |  |  |
| 7 | Camping car ID | CIN | VARCHAR | 17 |  | **☑** |
| 8 | Issue date | ISSUE\_DT | DATE | 3 |  | **☑** |

Table 10 - INSURANCE

Table 10 shows the insurance table that contains the data of each applied insurance to the camping car. It has IIN (Insurance ID as a primary key. CIN (Camping Car ID) and ISSUE\_DT (Issue Date) are the foreign key that reference the Camping Car ID and Issue Date from the camping car table. Due to the entity integrity, IIN (Insurance ID) should be unique and not null. In the aspect of the referential integrity, CIN (Camping Car ID) and ISSUE\_DT (Issue Date) should be updated and deleted casually when the referenced key is updated and deleted, because information of insurance only can exist when the camping car exists.

# **Choose**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) |
| 1 | Reservation ID | RID | VARCHAR | 20 | **☑** | **☑** |
| 2 | Accessory ID | ACSRY\_NO | VARCHAR | 10 | **☑** | **☑** |

Table 11 - CHOOSE

Table 11 shows the choose table, which contains the relationship between reservation and accessory. This table is needed due to resolve the numerous to numerous relations between the reservation and accessory table, which causes the problem of large required data size due to the duplication of tuple. It has RID (Reservation ID) and ACSRY\_NO (Accessory ID) as both primary and foreign keys that each reference Reservation ID from reservation table and Accessory ID from accessory table. In the aspect of entity integrity, RID (Reservation ID) and ACSRY\_NO (Accessory ID) should be unique and not null. In the aspect of the referential integrity, if the referenced keys are updated or deleted, both RID (Reservation ID) and ACSRY\_NO (Accessory ID) should be updated and deleted casually.

# **Apply**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) |
| 1 | Reservation ID | RID | VARCHAR | 20 | **☑** | **☑** |
| 2 | Event ID | EIN | INTEGER | 4 | **☑** | **☑** |

Table 12 - APPLY

Table 12 shows the apply table, which contains the information of applied events to the reservations. This table is needed due to resolve the numerous to numerous relations between the reservation and event table, which caused the problem of large required data size due to the duplication of tuple. If has RID (Reservation ID) and EIN (Event ID) as both primary and foreign keys that each reference Reservation ID from reservation table and Event ID from event table. In the aspect of entity integrity, RID (Reservation ID) and EIN (Event ID) should be unique and not null. In the aspect of referential integrity, , if the referenced keys are updated or deleted, both RID (Reservation ID) and EIN (Event ID) should be updated and deleted casually.

1. **Payment Type**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) |
| 1 | Payment type | PAY\_TP | INTEGER AUTO INCREMENT | 4 | **☑** |  |
| 2 | Payment type name | PAY\_NM | VARCHAR | 10 |  |  |

Table 13 - PAY\_TPYE

Table 13 shows the payment type table, which includes the information that is indexed with auto incremented numbers. This table is implemented due to identify the payment type whether it is paid by cash, card, or coupons. It has PAY\_TP (Payment Type) as a primary key, which follows the entity integrity. Therefore, it is unique and not null.

1. **Pay**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Name | Variable | Type | Size | Primary Key (PK) | Foreign Key (FK) |
| 1 | Payment ID | PAY\_ID | VARCHAR | 10 | **☑** |  |
| 2 | Payment type | PAY\_TP | INTEGER | 4 |  | **☑** |
| 3 | Payment amount | PAY\_AMT | FLOAT | 4 |  |  |
| 4 | Payment date | PAY\_DT | DATE | 3 |  |  |

Table 14 - PAY

Table 14 shows the pay table, which includes the information of payment. This table has PAY\_ID (Payment ID) as a primary key, and PAY\_TP (Payment Type) as a foreign key which references the Payment Type in payment type table. Due to the entity integrity, PAY\_ID (Payment ID) should be unique and not null. In the aspect of the referential integrity, PAY\_TP (Payment Type) should not be deleted until the referenced key is deleted. However, if the referenced key is updated, PAY\_TP (Payment Type) should updated casually.