Database Management Systems

MySQL - Integrity Control

Malay Bhattacharyya

Assistant Professor

Machine Intelligence Unit
and
Centre for Artificial Intelligence and Machine Learning
Indian Statistical Institute, Kolkata

March, 2020

1 Integrity Control

2 Basic Integrity Preservation
- Fundamentals
- Primary Key
- Foreign Key
- Nullity Check
- General Check

3 Problems

Basics

The term integrity in databases refers to the accuracy and consistency of data. The integrity control can set the following types of constraints on the data items:

- Basic integrity constraints
- Advanced integrity constraints

Basic integrity constraints

The basic integrity constraints are of following four types:

- Defining the `primary key` constraint
  – specified as `primary key` $(A_1, \ldots, A_k)$
- Defining the `foreign key` constraint
  – specified as `foreign key` $(A_p, \ldots, A_q)$ `references` r
- Defining the `null` constraint
  – specified as `not null`
- Defining the `check` constraint
  – specified as `check <predicate>`

Consider a relational schema

- Branch = $\langle \underline{branch\_id} : integer, branch\_name : string, branch\_city : string, assets : real \rangle$
- Customer = $\langle \underline{customer\_id} : integer, customer\_name : string, customer\_street : string, customer\_city : string, account\_number : integer \rangle$
- Loan = $\langle loan\_number : integer, branch\_name : string, amount : real \rangle$
- Borrower = $\langle customer\_name : string, loan\_number : integer \rangle$
- Account = $\langle \underline{account\_number} : integer, branch\_name : string, balance : real \rangle$
- Depositor = $\langle customer\_name : string, account\_number : integer \rangle$

Basic integrity constraints – primary key

The table Branch can be created with the following SQL query:

```
create table Branch(
 branch_id int(10) not null,
 branch_name varchar(30),
 branch_city varchar(30),
 assets float(20,2),
 primary key (branch_id)
);
```

Basic integrity constraints – foreign key

The table Customer can be created with the following SQL query:

```
create table Customer(
 customer_id int(20) not null,
 customer_name varchar(30),
 customer_street varchar(30),
 customer_city varchar(30),
 account_number int(20),
 primary key (customer_id),
 foreign key (account_number) references Account
);
```

Basic integrity constraints – null

The table Loan can be created with the following SQL query:

```
create table Loan(
  loan_number int(10) not null,
  branch_name varchar(30),
  amount float(15,2)
);
```

Basic integrity constraints – check

It can be ensured that a predicate on the attributes (say the amount is non-negative) must be satisfied by every tuple in the table Loan by writing an SQL query as follows:

```
 create table Loan(
  loan_number int(10) not null,
  branch_name varchar(30),
  amount float(15,2),
  check (amount >= 0)
);
```

Problems

1. Consider the following schema of an online code repository system like GitHub:
   - Contributor = $\langle contributor\_name : string, contributor\_id : integer \rangle$
   - Code-Group = $\langle \underline{contributor\_id} : integer, code\_group : string, count\_submissions : integer \rangle$

   Set the basic integrity constraints on this schema.