

Mysql 1

Introduction

- History
- Basic Terminology
- Relational Model versus Relational DBMS
- Domain
- Relation
- Relational Data Integrity

History

- E. F. Codd
 - A relational model of data for large shared data banks (1970)
- Set-Theory model (Childs, 1968)
- Goals of the RM (relational model)
 - high degree of data independence
 - theory of basic problems e.g. consistency, redundancy (normalization)
 - use of set-oriented data manipulation

Terminology

Common Term

Table

Record or Row

Field or Column

Number of columns

Number of rows

Identifier

Pool of legal values

Relational Term

Relation

Tuple

Attribute

Degree

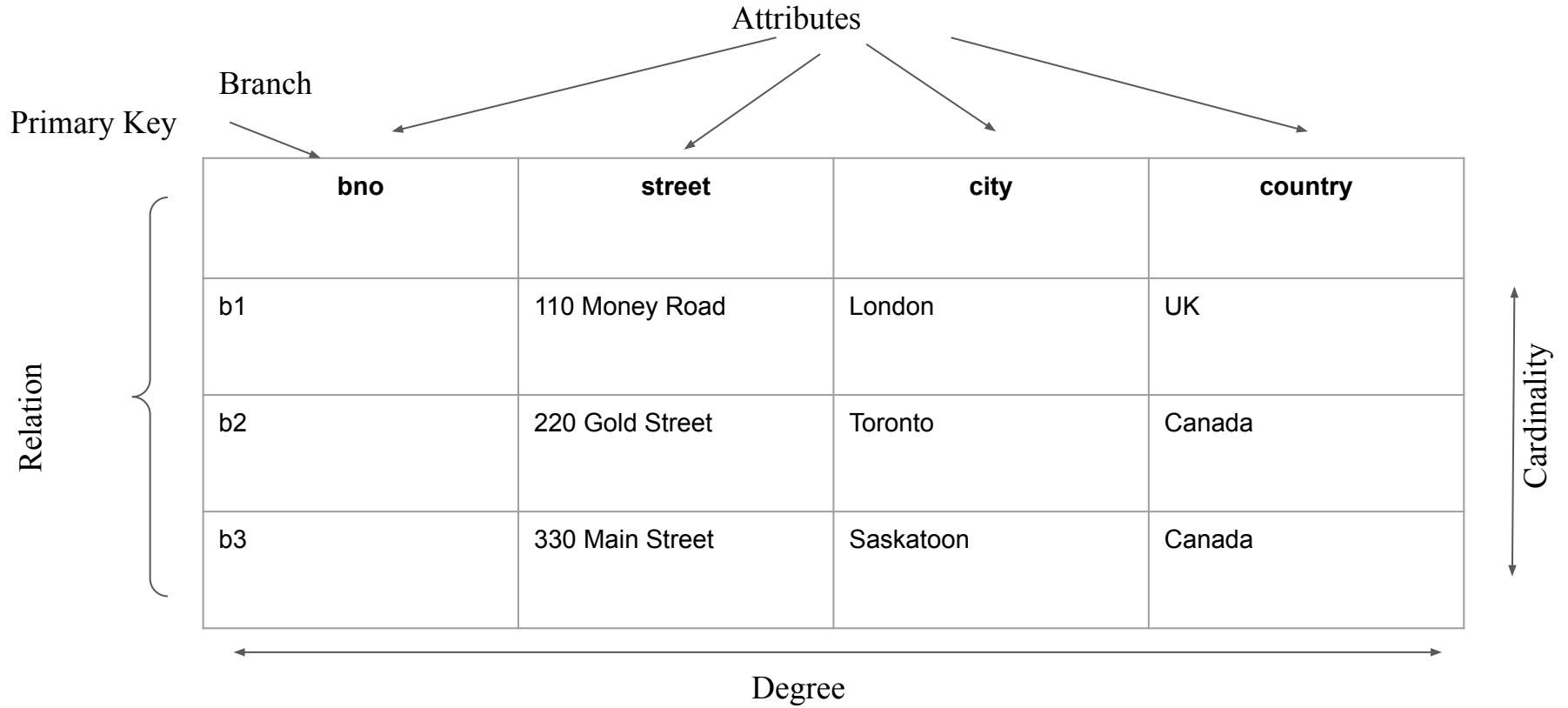
Cardinality

Primary Key

Domain

RM versus Relational DBMS

- There are differences between RM and relational DBMS
 - RM no duplicate tuples
 - Domains
 - etc...



Domain

- Pool of values (like a type)
- Values are atomic
- Domains have names
 - unique in database
 - are set of values
 - different domains can overlap
- e.g. {white, red, blue, green }

Relation 1

- Relations consist of
 - heading & body
- Heading
 - fixed set of attributes
 - e.g. $\{ \langle A1:D1 \rangle, \langle A2:D2 \rangle, \dots \langle AN:DN \rangle \}$
- Body
 - set of tuples
 - e.g. $\{ T1, T2, TN \}$
 - Tuple
 - $\{ \langle A1:V1 \rangle, \langle A2:V2 \rangle \dots \langle An:VN \rangle \}$

Relation 2

- Heading for relation S

- { bno, street, city, country }
- {
 - <bno : DOMAIN-BNO>
 - <street : DOMAIN-STREET>
 - <city : DOMAIN-CITY>
 - <country : DOMAIN-COUNTRY>
- }

Relation 3

- Tuple

- { b1, 110 Money Road, London, UK }

- {

- <bno : 'b1'>

- <street: '110 Money Road'>

- <city : 'London'>

- <country : 'UK'>

- }

Relation 4

- We interpret as humans the relation
- Difference between relation and table
 - relation \rightarrow abstract
 - relation \rightarrow linked to set theory
 - table \rightarrow concrete thing
 - table \rightarrow allows duplicate tuples

Relation 5

- Table -> suggest order
 - » tuple
 - » attributes

But there is no order of tuples or attributes!

Relation 6

- Degree (arity)
 - columns
 - are domains relations?
 - NO
 - relation dynamic
 - domain static
- Cardinality
 - rows

Operations

- Relation \rightarrow Relation
- Operations
 - Selection
 - Projection
 - Cartesian Product
 - Union
 - Intersection
 - Difference
 - Join
 - natural join
 - semi join
 - left outer join
 - Division

Properties of Relations

- Relation has a name that is distinct from other names
- Each cell of the relations contains only a single value
- Each attribute has a distinct name
- The values of an attribute are from the same domain
- The order of attributes is not important

Normalized Relations

- Normalization
 - A technique for producing a set of relations with desirable properties, given the data requirements of the enterprise
- Different types of NF - 1,2,3,4,5
- 1NF
 - A relation in which the intersection of each row and column contains only one value
- 2NF
 - Relation is in 1NF and non key attributes are fully dependent on key
- 3NF
 - Relation is in 2NF and all non-key attributes are not dependent on other non-key attribute

Relational Keys

- Super Key
 - An attribute or a set of attributes that uniquely identify a tuple within a relation
- Candidate Key
 - A super key such that no proper subset is a super key within the relation
- Primary key
 - The candidate key that is selected to identify tuples uniquely within the relation

Relational Data Integrity

- Candidate keys
 - uniqueness property
 - irreducibility property
- Why candidate keys?
 - Reference tuple
 - CMPT 353 - (c) Ralph Deters - 2025
 -
- Primary key
 - a special candidate key
- Foreign key
 - referential integrity (no unmatched foreign keys)

MySQL

docker-compose.yml

```
services:
mysql1:
  image: mysql:8.0
  container_name: db1
  volumes:
    - /Users/ralph/classes/353/w6/dbfiles:/var/lib/mysql
  environment:
    MYSQL_ROOT_PASSWORD: admin_xxx
    MYSQL_DATABASE: my_database
    MYSQL_USER: user1
    MYSQL_PASSWORD: user1_xxx
```

```
node1:
  build: .
  container_name: nodejs1
  ports:
    - "80:8080"
  volumes:
    - /Users/ralph/classes/353/w6:/usr/src/app
  depends_on:
    mysql1:
      condition: service_started
  stdin_open: true
  tty: true
```

```
docker exec -it <container_name_or_id> mysql -u <username> -p
```

```
docker exec -it db1 bash
```

```
mysql -u root -p
```

Create table

```
show databases;
```

```
create database db1;
```

```
drop database db1;
```

```
create database db1;
```

```
use db1;
```

```
CREATE TABLE mytable  
(  
  id int unsigned NOT NULL auto_increment,  
  username varchar(100) NOT NULL,  
  email varchar(100) NOT NULL,  
  PRIMARY KEY (id)  
);
```



```
describe mytable;
```

```
ALTER TABLE mytable ADD COLUMN phone_number VARCHAR(20);
```

```
ALTER TABLE mytable drop COLUMN phone_number;
```

Insert

```
INSERT INTO mytable ( username, email ) VALUES ( "myuser",  
"myuser@example.com" );
```

Select

```
SELECT C1, C2, C3 FROM table where C1 = "1";
```

```
select * from mytable;
```

Update

UPDATE table_name SET column1 = value1, column2 = value2, column3 = value3, ... WHERE condition;

UPDATE mytable SET username="myuser_1" WHERE id=1;