COM 745

Introduction to relational databases

Dr. Joe Rafferty

Ulster University

1

Relational Databases

Introduction

- Relational database management systems (RDBMS) were proposed in the 1970s by E.F. Codd
- This paradigm dominated database design until the turn of the century
- Examples of RDBMS include:
    - ***MySQL***
    - MS SQL
    - MS Access
    - Oracle SQL

Ulster University

2

Relational Databases
Introduction

- Relational database management systems (RDBMS) have a number of defining characteristics
  - Data in a RDBMS is organised through the “Relational Model”
  - Support Structured Query Language (SQL) for interacting with a database
  - Support relationships between tables
  - Typically support ACID transactions

Ulster University

3

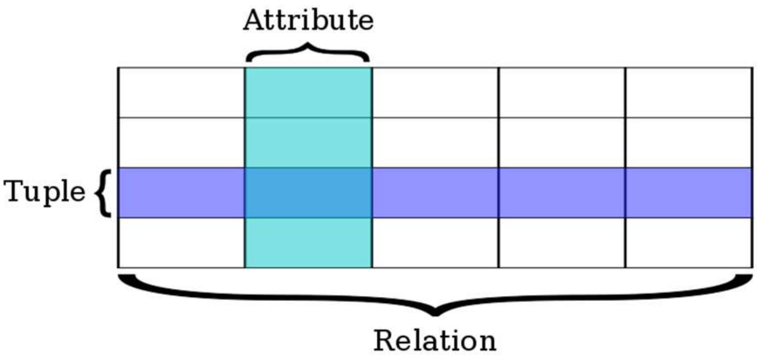
Relational Databases
Relational Model

- Relational models are essentially tables within a database
- Tables generally represent one entity type, such as student/customer records
- These tables consist of rows and columns
  - Rows may be referred to as Tuples or Records
  - Columns may be referred to as Fields or Attributes
  - Tables may be referred to as Relations or Base Revlar*

Ulster University

4

Relational Databases
Relational Model



5

Relational Databases
Relational Model

Customer ID	Tax ID	Name	Address	[More fields...]
1234567890	555-5512222	Munmun	323 Broadway	...
2223344556	555-5523232	Wile E.	1200 Main Street	...
3334445563	555-5533323	Ekta	871 1st Street	...
4232342432	555-5325523	E. F. Codd	123 It Way	...

6

Relational Databases
Relational Model

- The relational model can have a number of keys to identify records, these include:
    - Primary Keys (PK)
    - Composite Keys (CK)
    - Foreign Keys (FK)

Ulster University

7

Relational Databases
Relational Model

- Primary Keys
    - A field offering a unique identifier
    - Used to identify a unique tuple
    - May be an auto-incrementing numeric value

Store ID	Purchase Location
1	Los Angeles
2	New York
3	San Francisco

Ulster University

8

Relational Databases
Relational Model

- Composite Keys
  - Used to identify a unique tuple
  - A combination of two or more columns
  - May employ auto incrementing on one, some or all columns involved

Customer ID	Store ID	Purchase Location
1	1	Los Angeles
1	3	San Francisco
2	1	Los Angeles
3	2	New York
4	3	San Francisco

Ulster University

9

Relational Databases
Relational Model

- Foreign Keys
  - A field in one table that uniquely identifies a key in another table
  - Facilitates selection of records across tables
  - Enforces referential integrity
  - May be an auto-incrementing numeric value

Ulster University

10

Relational Databases

Relational Model

- Foreign Keys
  - A field in one table that uniquely identifies a key in another table
  - Facilitates selection of records across tables
  - Enforces referential integrity
    - A FK may only contain values from a specified column in a foreign table or a null value – providing assurance that related information exists
    - When deleting a record containing a FK the RDBMS may delete the record in the other table or prevent deletion

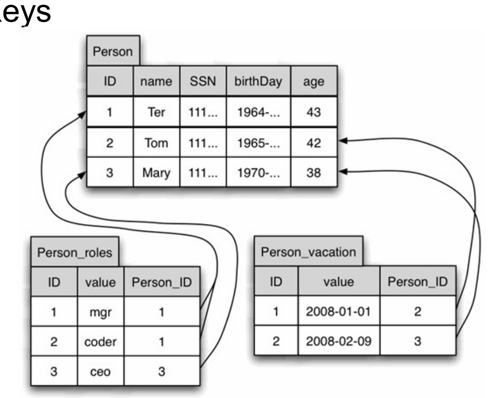
Ulster University

11

Relational Databases

Relational Model

- Foreign Keys



Ulster University

12

Relational Databases
Normalization

- Relations/Tables should be carefully defined
- The databases should be “Normalized”
- Normalization is the process of reorganising attributes into tables with the goal of minimising redundancy
  - Increases storage efficiency
  - May have performance implications
- Normalization may take a number of forms

Ulster University

13

Relational Databases
Normalization

- An example of normalization in a sale database
  - Storing sales records and product records in separate tables
  - Product records are stored within unique IDs
  - Records in the sales table reference items sold through referencing a unique ID

Ulster University

14

Relational Databases
Normalization

- Three different levels of normalization
  - 1NF – First Normal Form
  - 2NF – Second Normal Form
  - 3NF – Third Normal Form

- A DB is described as normalized when it satisfies the 3NF

Ulster University

15

Relational Databases
Normalization – 1NF

- 1NF enforces the following criteria
  - Eliminate repetition in individual tables
  - Create a separate table for each set of related data/entity type
  - Identify each record with a primary key
  - Contains only atomic values

Ulster University

16

Relational Databases
Normalization – 1NF

- Given the requirement to store multiple phone numbers
- A violates the requirement as telephone number may only contain one atomic value

Customer ID	First Name	Surname	Telephone Number
123	Robert	Ingram	555-861-2025
456	Jane	Wright	555-403-1659 555-776-4100
789	Maria	Fernandez	555-808-9633

A

- B satisfies the 1NF but not more complex forms

Customer ID	First Name	Surname	Telephone Number
123	Robert	Ingram	555-861-2025
456	Jane	Wright	555-403-1659
456	Jane	Wright	555-776-4100
789	Maria	Fernandez	555-808-9633

B

Ulster University

17

Relational Databases
Normalization – 2NF

- 2NF enforces the following criteria
  - Satisfy criteria expressed in 1NF
  - Non-key attributes are fully functionally dependent on the primary key

Ulster University

18

Relational Databases
Normalization – 2NF

- A presents a table using a CK that is not normalised.
- A violates the requirement as Purchase Location only depends on store ID which is only an element of the PK
- B satisfies the 2NF

Customer ID	Store ID	Purchase Location
1	1	Los Angeles
1	3	San Francisco
2	1	Los Angeles
3	2	New York
4	3	San Francisco

A

Customer ID	Store ID
1	1
1	3
2	1
3	2
4	3

Store ID	Purchase Location
1	Los Angeles
2	New York
3	San Francisco

B

Ulster University

19

Relational Databases
Normalization – 3NF

- 3NF enforces the following criteria
  - Satisfy criteria expressed in 2NF
  - “Every non-prime attribute of R is non-transitively dependent on every key of R”
- More simply put:
  "Every non-key attribute must provide a fact about the key, the whole key, and nothing but the key.“ – Bill Kent

Ulster University

20

Relational Databases
Normalization – 3NF

- A is not in the 3NF, Book ID determines Genre ID and Genre ID determines Genre Type. This represents a transitive functional dependency.

Book ID	Genre ID	Genre Type	Price
1	1	Gardening	25.99
2	2	Sports	14.99
3	1	Gardening	10.00
4	3	Travel	12.99
5	2	Sports	17.99

A

- B is brought to the 3rd normal form by splitting into two tables

Book ID	Genre ID	Price
1	1	25.99
2	2	14.99
3	1	10.00
4	3	12.99
5	2	17.99

Genre ID	Genre Type
1	Gardening
2	Sports
3	Travel

B

Ulster University

21

Relational Databases
Entity-Relationship Modelling

- Mapping between tables may take a number of forms:
    - 1 to 1 – e.g one person may have one social security number
    - 1 to Many - e.g one person may have many bank accounts
    - Many to Many – e.g. books may have many authors, and vice versa

Book ID	Genre ID	Price
1	1	25.99
2	2	14.99
3	1	10.00
4	3	12.99
5	2	17.99

Genre ID	Genre Type
1	Gardening
2	Sports
3	Travel

Ulster University

22

Relational Databases
ACID transactions

- RDBMS typically support ACID transactions
- ACID transactions offer the following properties
  - Atomicity – an entire operation is performed or nothing is
  - Consistency – any transaction will bring the DB from one valid state to another
  - Isolation – concurrent transactions would not interfere with one another
  - Durability – once a transaction has been committed it will remain even in the event of power loss or and error

Ulster University

23

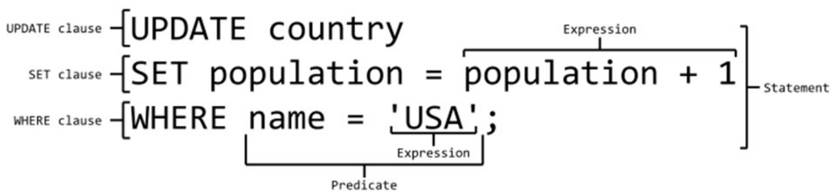
Relational Databases
SQL

- SQL is the primary mechanism for interacting with RDBMS
- SQL is subdivided into a number of language elements, including:
  - Clauses - components of statements and queries
  - Expressions – used to evaluate a value
  - Predicates – conditions to be evaluated within SQL
  - Queries – Retrieve data based on criteria
  - Statements – instructions which may have an effect on data or DB schema

Ulster University

24

Relational Databases
SQL



25

Relational Databases
SQL

- Examples of clauses include:
  - SELECT – select data from a database
  - INSERT – insert data into a database
  - UPDATE – update records in a database
  - DELETE – delete records in a database
  - WHERE – used to restrict the operation of a statement
  - LIMIT – used to limit the number of records returned by a resultset

26

Relational Databases
SQL

- Examples of clauses include
  - JOIN – used to combine rows from 2+ tables
  - DISTINCT – used to remove duplicates from the result set produced by a select statement
  - ORDER BY – used to order the result set of a query by a specified column list
  - GROUP BY – used in a select statement to collect data across multiple records and group the results

Ulster University

27

Relational Databases
SQL

- Examples of clauses include
  - HAVING – similar to where but applied to aggregate functions, such as GROUP BY
  - UNION - combine results of 2+ select statements without returning duplicate rows

Ulster University

28

Relational Databases
SQL

- The following is the structure for a statement that creates a table within SQL

```
CREATE TABLE table_name
(
column_name1 data_type(size),
column_name2 data_type(size),
column_name3 data_type(size),
PRIMARY KEY (column_name)
);
```

Ulster University

29

Relational Databases
SQL – Data types

- SQL supports a number of data types including:
  - CHAR – a number of characters up to 255; char (30)
  - VARCHAR – a number of characters up to 65,535; varchar (300)
  - TEXT – strings/text cannot be ordered or used in a where clause, in most cases
  - BINARY – binary values
  - BLOB – binary strings

Ulster University

30

Relational Databases
SQL – Data types

- SQL supports a number of data types including:
  - INT – whole numbers, 4 bytes, signed or unsigned
    - -2147483648 to 2147483647
    - 0 to 4294967295
  - BIGINT – whole numbers, 8 bytes, signed or unsigned
    - -9223372036854775808 to 9223372036854775807
    - 0 to 18446744073709551615

- These may vary with RDBMS implementation
- Correct selection and specification is imperative
- Incorrect specification has performance implications

Ulster University

31

Relational Databases
Storage hierarchy

There is a storage hierarchy present within Relational databases:

1. Database server
2. Schemas/Databases*
3. Tables
4. Relations

Ulster University

32

Relational Databases
Schema

Schemas/databases are used to store related tables.

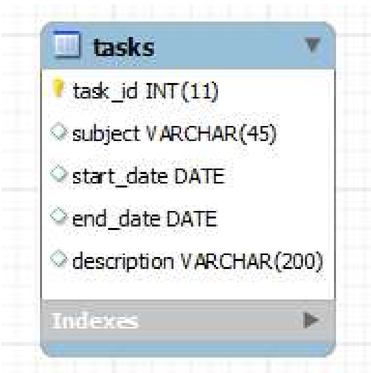
The SQL command to create a schema is:

```
CREATE SCHEMA `example_schema` ;
```

Once a schema is in place, tables may be created.

33

Relational Databases
Tables



34

Relational Databases
Tables

- To create this table the following statement is used:

```
CREATE TABLE IF NOT EXISTS example_schema.tasks (
  task_id INT(11) NOT NULL AUTO_INCREMENT,
  subject VARCHAR(45) DEFAULT NULL,
  start_date DATE DEFAULT NULL,
  end_date DATE DEFAULT NULL,
  description VARCHAR(200) DEFAULT NULL,
  PRIMARY KEY (task_id)
) ENGINE=InnoDB
```

Ulster University

35

Relational Databases
Relations/Records

- After a table is created a relation/record may inserted
- To insert a relation/record the following SQL is used

```
INSERT INTO `example_schema`.`tasks`
(`subject`, `start_date`, `end_date`, `description`)
VALUES
('SubjectName', '1970-01-01', '1970-01-01', 'A lengthy
description');
```

Note the missing field

Ulster University

36

Relational Databases
Relations/Records

- After a relation is inserted it may be Selected as part of a result set

```
SELECT * FROM example_schema.tasks
WHERE task_id > 0 ORDER BY subject;
```

task_id	subject	start_date	end_date	description
2	ASubjectName	1970-01-01	1970-02-01	Another desc
1	SubjectName	1970-01-01	1970-01-01	A lengthy de...
NULL	NULL	NULL	NULL	

Ulster University

37

Relational Databases
Relations/Records

- Records present within an database may be updated through the following command

```
UPDATE example_schema.tasks
SET subject="UpdatedSubject" WHERE task_id = 2;
```

task_id	subject	start_date	end_date	description
1	SubjectName	1970-01-01	1970-01-01	A lengthy de...
2	UpdatedSubject	1970-01-01	1970-02-01	Another desc
NULL	NULL	NULL	NULL	NULL

Ulster University

38

Relational Databases
Relations/Records

- Records may be deleted with the DELETE command

```
DELETE FROM example_schema.tasks
WHERE task_id >1;
```

	1	SubjectN...	1970-01-01	1970-01-01	A lengthy de...
*	NULL	NULL	NULL	NULL	NULL

Ulster University

39

Relational Databases
Advanced topics

Join – This allows querying across tables to produce a record set/Revlar

Consider the tasks table coexisting with the following "task_operator" table

Column	Type	Default Value	Nullable
operator_id	int(11)		NO
task_id	int(11)		YES
operator_name	varchar(450)		YES

Ulster University

40

Relational Databases
Advanced topics

The task operator table contains the following

	operator_id	task_id	operator_name
	1	1	Anon
	2	2	Bnon
	3	3	Cnon
	4	1	Dnon
▸*		NULL	NULL

41

Relational Databases
Advanced topics

Using join it is possible to find the operators associated with a task simply by specifying the task name.

```
SELECT T1.operator_name FROM example_schema.task_operator T1
INNER JOIN example_schema.tasks T2 ON T2.task_id = T1.task_id
WHERE T2.subject = "SubjectName";
```

	operator_name
	Anon
	Dnon

42

Relational Databases
Advanced topics

DISTINCT – when distinct is used in a clause it will select remove duplicates.

E.G `SELECT DISTINCT(SubjectName) FROM example_schema.tasks;`

Stored Procedures – SQL statements stored within a DB, removes a layer of logic from consuming applications.

Triggers – A procedure or function stored within a DB that activates the a certain condition is met.

Ulster University

43

Relational Databases
Use cases

- Use for relational records where
  - Single / low numbers of related instances will be stored*
  - Applications where ACID consistency is required
- Examples include
  - User profiles, e.g. name + login info
  - Reliable storage of transactional records, such as financial logs**
  - Address books

Ulster University

44

Relational Databases
Use cases

- When not to use a relational database
  - High volume data, such as logs that do not require ACID consistency
  - Storage of binary data; such as images, videos and objects*
  - Storage of XML**

Ulster University

45

Relational Databases

- Break time
- Practical aspect to follow

Ulster University

46

47