Maharishi International University
Master's in Software Development

CS418:
Databases & Software Development

Lesson 2

The Relational Model and SQL

Wholeness

- This lecture focuses on studying the underpinning concepts behind Relational Databases, which is the Relational model.
- We will also learn the language for working with data in a Relational database - SQL. *The Structured Query language (SQL) is a standard across any RDBMS. This is akin to how the Unified field is the basis for all of life.*

Lesson 2 Objectives

- The origins of the Relational model.
- The terminology of the Relational model.
- How tables are used to represent data.
- The connection between mathematical relations and relations in the relational model.
- Properties of database relations.
- How to identify primary and foreign keys.
- The meaning of entity integrity and referential integrity.
- The purpose and advantages of views in relational systems.
- …

2.1 Introduction to the Relational model

- The Relational Database Management System (RDBMS) has been the dominant database technology in the enterprise for much of the last 50 years.
- It represents the second generation of DBMSs and is based on the relational data model proposed by E. F. Codd (1970).
- In the relational model, all data is logically structured within **relations (tables)**. Each relation has a name and is made up of named **attributes (columns)** of data. Each **tuple (row)** contains one value per attribute. A great strength of the relational model is this simple logical structure.

2.2 Brief history

- The Relational model was first proposed by E. F. Codd in his seminal paper “A relational model of data for large shared data banks” in 1970.
- Based on Codd’s paper, IBM's San Jose Research lab implemented a prototype relational DBMS named, System R, which included the development of a structured query language, SQL – now the ISO/de facto standard language for RDBMSs.

2.2 Brief history

- During the late 1970’s and the 1980’s various commercial RDBMSs were produced – beginning with IBM’s DB2 and Oracle from Oracle Corporation.
- Today, there are several hundred RDBMSs for both mainframe computers and PC environments.

2.3 Terminology

- The relational model is based on the mathematical concept of a **relation**. The concept is derived from the mathematical topic of set theory and predicate logic.
- Relation: A relation is a table of columns and rows.
- Attribute: An attribute is a named column of a relation.
- Domain: A domain represents the set of allowable values for one or more attributes. Every attribute in a relation is defined on a domain.
- Tuple: A tuple is a row of a relation. The elements of a relation are the rows or tuples in the table.

2.3 Terminology

- Degree: The degree of a relation is the number of attributes it contains. A relation with only one attribute would have degree one and be called a unary relation or one-tuple. A relation with two attributes is called binary, one with three attributes is called ternary, and after that the term n-ary is usually used.
- Cardinality: The cardinality of a relation is the number of tuples it contains. This changes as tuples are added or deleted.

2.3 Terminology

- Relational Database: a collection of **normalized** relations with distinct relation names. A relational database consists of relations that are appropriately structured. We refer to this appropriateness as *normalization*. We will study Normalization in further detail in Lessons 3 and 4.

FORMAL TERMS	ALTERNATIVE 1	ALTERNATIVE 2
Relation	Table	File
Tuple	Row	Record
Attribute	Column	Field

Alternative terminology for relational model terms.

Example

Figure 2.1

Attributes

Branch

Relation

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Cardinality

Degree

Primary key

Foreign key

Staff

Relation

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

Attribute	Domain Name	Meaning	Domain Definition
branchNo	BranchNumbers	The set of all possible branch numbers	character: size 4, range B001–B999
street	StreetNames	The set of all street names in Britain	character: size 25
city	CityNames	The set of all city names in Britain	character: size 15
postcode	Postcodes	The set of all postcodes in Britain	character: size 8
sex	Sex	The sex of a person	character: size 1, value M or F
DOB	DatesOfBirth	Possible values of staff birth dates	date, range from 1-Jan-20, format dd-mmm-yy
salary	Salaries	Possible values of staff salaries	monetary: 7 digits, range 6000.00–40000.00

2.4 Mathematical relations

To understand the true meaning of the term *relation*, we have to review some concepts from mathematics. Suppose that we have two sets, D_1 and D_2, where $D_1 = \{2, 4\}$ and $D_2 = \{1, 3, 5\}$. The **Cartesian product** of these two sets, written $D_1 \times D_2$, is the set of all ordered pairs such that the first element is a member of D_1 and the second element is a member of D_2. An alternative way of expressing this is to find all combinations of elements with the first from D_1 and the second from D_2. In our case, we have:

$$D_1 \times D_2 = \{(2, 1), (2, 3), (2, 5), (4, 1), (4, 3), (4, 5)\}$$

Any subset of this Cartesian product is a relation. For example, we could produce a relation R such that:

$$R = \{(2, 1), (4, 1)\}$$

We may specify which ordered pairs will be in the relation by giving some condition for their selection. For example, if we observe that R includes all those ordered pairs in which the second element is 1, then we could write R as:

$$R = \{(x, y) \mid x \in D_1, y \in D_2, \text{ and } y = 1\}$$

Using these same sets, we could form another relation S in which the first element is always twice the second. Thus, we could write S as:

$$S = \{(x, y) \mid x \in D_1, y \in D_2, \text{ and } x = 2y\}$$

or, in this instance,

$$S = \{(2, 1)\}$$

as there is only one ordered pair in the Cartesian product that satisfies this condi-

2.5 Relational keys

- Primary key: A chosen attribute, or set of attributes, that uniquely identifies a tuple within a relation.
- Foreign key: An attribute, or set of attributes, within one relation that matches the primary key of some other relation.

2.5 Relational keys

- When an attribute appears in more than one relation, its appearance usually represents a relationship between tuples of the two relations. For example, in Figure 2.1 above, the inclusion of branchNo in both the Branch and Staff relations is quite deliberate and links each branch to the details of staff working at that branch. In the Branch relation, branchNo is the primary key. However, in the Staff relation, the branchNo attribute exists to match staff to the branch office they work in. In the Staff relation, branchNo is a foreign key.

2.6 Representing relational database schema

- The common convention for representing a relation schema is to give the name of the relation followed by the attribute names in parentheses. Normally, the primary key is underlined. For example, the relational schema for part of the *DreamHome* case study is:

Branch	(branchNo, street, city, postcode)
Staff	(staffNo, fName, lName, position, sex, DOB, salary, branchNo)
PropertyForRent	(propertyNo, street, city, postcode, type, rooms, rent, ownerNo, staffNo, branchNo)
Client	(clientNo, fName, lName, telNo, prefType, maxRent, eMail)
PrivateOwner	(ownerNo, fName, lName, address, telNo, eMail, password)
Viewing	(clientNo, propertyNo, viewDate, comment)
Registration	(clientNo, branchNo, staffNo, dateJoined)

2.6 Representing relational database schema

- The *conceptual model,* or *conceptual schema,* is the set of all such schemas for the database.

2.6 Representing relational database schema

- Instance of the database schema for the *DreamHome* rental database:

Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

PropertyForRent

propertyNo	street	city	postcode	type	rooms	rent	ownerNo	staffNo	branchNo
PA14	16 Holhead	Aberdeen	AB7 5SU	House	6	650	CO46	SA9	B007
PL94	6 Argyll St	London	NW2	Flat	4	400	CO87	SL41	B005
PG4	6 Lawrence St	Glasgow	G11 9QX	Flat	3	350	CO40		B003
PG36	2 Manor Rd	Glasgow	G32 4QX	Flat	3	375	CO93	SG37	B003
PG21	18 Dale Rd	Glasgow	G12	House	5	600	CO87	SG37	B003
PG16	5 Novar Dr	Glasgow	G12 9AX	Flat	4	450	CO93	SG14	B003

2.6 Representing relational database schema

- Instance of the database schema for the *DreamHome* rental database cont'd:

Fig. 2.2

Client

clientNo	fName	lName	telNo	prefType	maxRent	eMail
CR76	John	Kay	0207-774-5632	Flat	425	john.kay@gmail.com
CR56	Aline	Stewart	0141-848-1825	Flat	350	astewart@hotmail.com
CR74	Mike	Ritchie	01475-392178	House	750	mritchie01@yahoo.co.uk
CR62	Mary	Tregear	01224-196720	Flat	600	maryt@hotmail.co.uk

PrivateOwner

ownerNo	fName	lName	address	telNo	eMail	password
CO46	Joe	Keogh	2 Fergus Dr, Aberdeen AB2 7SX	01224-861212	jkeogh@lhh.com	********
CO87	Carol	Farrel	6 Achray St, Glasgow G32 9DX	0141-357-7419	cfarrel@gmail.com	********
CO40	Tina	Murphy	63 Well St, Glasgow G42	0141-943-1728	tinam@hotmail.com	********
CO93	Tony	Shaw	12 Park Pl, Glasgow G4 0QR	0141-225-7025	tony.shaw@ark.com	********

Viewing

clientNo	propertyNo	viewDate	comment
CR56	PA14	24-May-13	too small
CR76	PG4	20-Apr-13	too remote
CR56	PG4	26-May-13	
CR62	PA14	14-May-13	no dining room
CR56	PG36	28-Apr-13	

Registration

clientNo	branchNo	staffNo	dateJoined
CR76	B005	SL41	2-Jan-13
CR56	B003	SG37	11-Apr-12
CR74	B003	SG37	16-Nov-11
CR62	B007	SA9	7-Mar-12

2.7 Integrity Constraints

- In addition to the structural part, a data model such as the relational data model, has two other parts: a manipulative part, defining the types of operation that are allowed on the data, and a set of integrity constraints, which ensure that the data is accurate.
- Because every attribute has an associated domain, there are constraints (called domain constraints) that form restrictions on the set of values allowed for the attribute.

2.7 Integrity Constraints

- In addition to the structural part, a data model such as the relational data model, has two other parts: a manipulative part, defining the types of operation that are allowed on the data, and a set of integrity constraints, which ensure that the data is accurate.
- Because every attribute has an associated domain, there are constraints (called domain constraints) that form restrictions on the set of values allowed for the attribute.

2.7 Integrity Constraints

- The relational model has two principal Integrity rules which govern constraints or restrictions that apply to all instances of the database. These are known as **entity integrity** and **referential integrity**. We will learn about these shortly.
- The concept of Nulls:
  – Null represents a value for an attribute that is currently unknown or is not applicable for the given tuple. e.g. in the *Viewing* relation shown in Figure 2.2 the *comment* attribute may be undefined until the potential renter has visited the property.

2.7 Integrity Constraints

- **Entity integrity:** means an attribute that is a primary key or part of a primary cannot be null.
- R**eferential integrity:** means if a foreign key exists in a relation, either the foreign key value must match a primary key value of some tuple in its home relation or the foreign key value must be wholly null.
  - For example, branchNo in the Staff relation is a foreign key targeting the branchNo attribute in the home relation, Branch. It should not be possible to create a staff record with branch number B025, for example, unless there is already a record for branch number B025 in the Branch relation. However, we should be able to create a new staff record with a null branch number to allow for the situation where a new member of staff has joined the company but has not yet been assigned to a particular branch office.

2.8 Views

- **Base relation:** A named relation corresponding to an entity in the conceptual schema, whose tuples are physically stored in the database.
- **View:** The dynamic result of one or more relational operations operating on the base relations to produce another relation. A view is a virtual relation that does not necessarily exist in the database but can be produced upon request by a particular user, at the time of request.

2.8 Views

- A view is a relation that appears to the user to exist, can be manipulated as if it were a base relation, but does not necessarily exist in storage in the sense that the base relations do (although its definition is stored in the system catalog). The contents of a view are defined as a query on one or more base relations. Any operations on the view are automatically translated into operations on the relations from which it is derived. Views are dynamic, meaning that changes made to the base relations that affect the view are immediately reflected in the view.

2.8 Views

- A view is a relation that appears to the user to exist, can be manipulated as if it were a base relation, but does not necessarily exist in storage in the sense that the base relations do (although its definition is stored in the system catalog). The contents of a view are defined as a query on one or more base relations. Any operations on the view are automatically translated into operations on the relations from which it is derived. Views are dynamic, meaning that changes made to the base relations that affect the view are immediately reflected in the view.

2.8 Views

- What is the purpose of Views?
  - It provides a powerful and flexible security mechanism by hiding parts of the database from certain users. Users are not aware of the existence of any attributes or tuples that are missing from the view.
  - It permits users to access data in a way that is customized to their needs, so that the same data can be seen by different users in different ways, at the same time.
- A view should be designed to support the external model that the user finds familiar

Conclusion

- This first part of lesson 2 has given you an overview of the Relational model.
- Next, we will study SQL, including its DML and DDL parts.

Connecting the Parts of Knowledge With the Wholeness of Knowledge

Overview and Introduction Relational Model and RDBMS

1. A database Management systems that is built on the Relational model is said to be a Relational Database Management system (RDBMS)
2. A Relation is represented in an RDBMS as a table.

---

3. **Transcendental consciousness** is the underlying basis of all levels of creation.
4. **Impulses within the Transcendental Field:** The well-designed Database which forms the basis of an elegant software solution, arises as an impulse of the Transcendental Field.
5. **Wholeness moving within itself**: In Unity Consciousness, one directly perceives that all expressions and levels of creation are nothing more than one's own Self pure consciousness.

Maharishi International University
Master's in Software Development

CS418: Databases & Software Development