

Relational Data Model

- **Dr. E. F. Codd** proposed the relational model for database systems in **1970**.
- It is the **basis** for the relational database management system (**RDBMS**), the most common DBMS used today
- The relational model **consists of the following**:
 - A collection of **relations**
 - each **relation** consists of a set of **attributes** and describes an object (or **Entity**) class
 - Data **integrity** for accuracy and consistency
 - **entity** integrity and **referential** integrity
 - **Set of operators** to act on the relations
 - **relational algebra**, **relational calculus** (**SQL** is a form of relational calculus)

- **Domain**
 - a set of **atomic (indivisible) values**
 - are pools of values from which the actual values appearing in attributes are drawn
 - specify
 - **name**
 - **data type**
 - **data format**
- **Examples:**
 - **gender** domain
 - one character string with allowable values of M, m, F, f
 - **name** domain
 - twenty-character string
 - **credit_limit** domain
 - money in the range \$1,000 to \$99,999

▪ Relation (table)

- a named set of attributes, consists of **two parts: heading** and **body**

▪ Relation Heading

- Also called a **Relation Schema** consists of a fixed set of attributes
 - $R (A_1, A_2, \dots, A_n)$
 - R = relation name, A_i = attribute i
 - **Customer relation heading or schema:** CUSTOMER (custno, custname, custadd, credlimit)
 - Each attribute corresponds to one underlying **domain**:
 - $\text{dom}(\text{custno}) = \text{customer_number}$
 - $\text{dom}(\text{custname}) = \text{name}$
 - $\text{dom}(\text{custadd}) = \text{address}$
 - $\text{dom}(\text{credlimit}) = \text{credit_limit}$

Diagrammatic
Representation

custno	custname	custadd	credlimit
--------	----------	---------	-----------

■ Relation Body

- Also called **Relation Instance** (state of the relation at any point in time)
 - $r(R) = \{t_1, t_2, t_3, \dots, t_m\}$
 - consists of a time-varying set of **tuples**, **$t_1, t_2, t_3 \dots t_m$**
- Consists of **records or rows**
- In the tabular representation:

Customer

	custno	custname	custadd	credlimit
t1 →	SMI13	SMITH	Wide Rd, OnePlace, 1111	2000
t2 →	JON44	JONES	Narrow St, Somewhere 2222	10000
t3 →	BRO23	BROWN	Here Rd, Lost, 3333	10000

Relation heading

Relation body

- **No duplicate** tuples
 - by definition sets do not contain duplicate elements
 - hence tuples are unique
- Tuples are **unordered** within a relation
 - by definition sets are not ordered
 - hence tuples can **only** be accessed by content
- **No ordering of attributes** within a tuple
 - by definition sets are not ordered
- **No multivalued attributes** in a relation
 - an entry at an intersection of each row and column is atomic (**single-valued**)

How does the model ensure this?



Entity Integrity (constraint): Primary key in a table must be unique and no part of the key is NULL

KEY constraints: Constraints are conditions that must hold true for all valid tuples

Summary of Keys

- **Super key**
 - Any key that uniquely identifies **each row in an entity**
- **Candidate key**
 - Minimal super key
- **Primary key**
 - Candidate key (chosen) to uniquely identify all other attributes in a given row
- **Secondary key**
 - Used only for data retrieval, cannot uniquely identify each row of the table
- **Composite key**
 - Key composed of more than one attribute
- **Key attribute**
 - Any attribute that is part of a key
- **Foreign key**
 - Values must match a primary key in a referenced (parent) table or be null

Key Examples

- Key Fields

- Super key
- Candidate key
- Primary key
- Foreign key
- Secondary key

COURSE Table

COURSE-ID	COURSE-DESCRIPTION	NUMBER-OF-CREDITS
CHM112	General Chemistry I	5
CSC151	Computer Science I	3
ENG101	English Composition	3
MKT212	Marketing Management	3

GRADE Table

STUDENT-NUMBER	COURSE-ID	GRADE
1035	CSC151	B
1035	MKT212	A
1035	ENG101	B
1035	CHM112	A
3397	ENG101	A
3397	MKT212	C
3397	CSC151	B
4070	CSC151	B
4070	CHM112	C

Another example:

Table Name: **EMP**

EMPNO	ENAME	JOB	DEPTNO
7839	KING	PRESIDENT	10
7698	BLAKE	MANAGER	30
7782	CLARK	MANAGER	10
7566	JONES	MANAGER	20

Table Name: **DEPT**

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Primary key

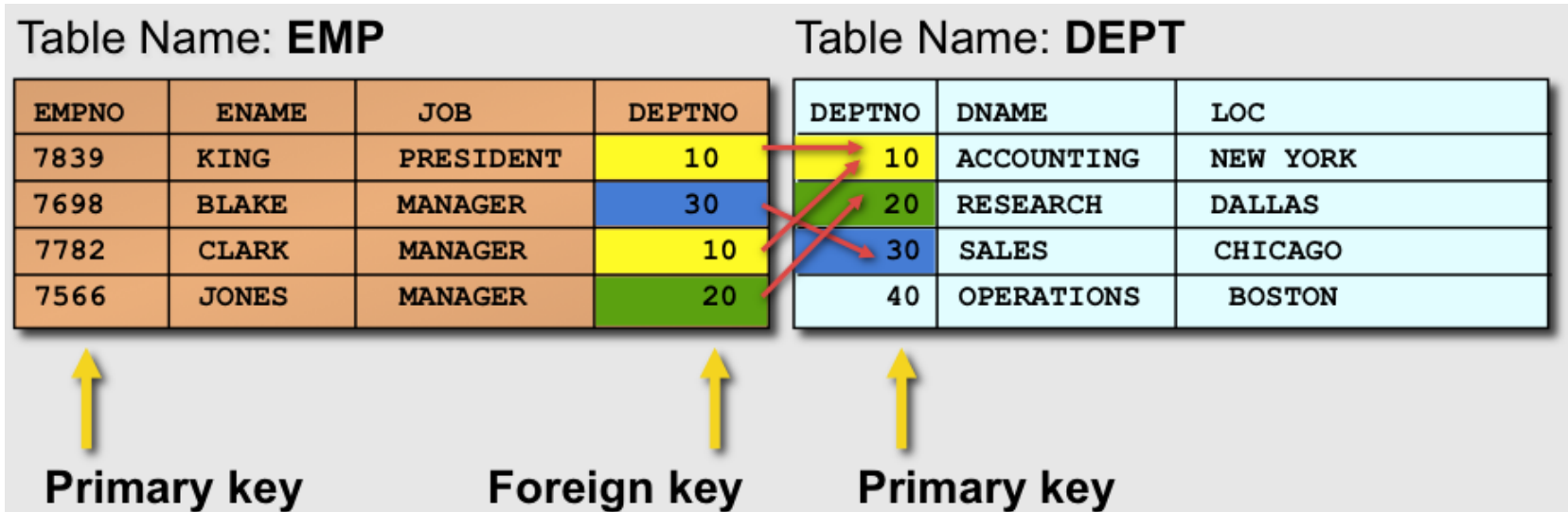
Foreign key

Primary key

- Primary key in EMP table?
- Primary key in DEPT table?
- Foreign key in EMP table?

Relating Multiple Tables

- Each row of data in a table is uniquely identified by a primary key (PK).
- You can logically relate data from multiple tables using foreign keys (FK).



- Referential integrity** (constraint): FK values in a table be either a valid value coming from another table or be NULL; DBMS always check if a valid FK value exists in FK all entries of a table.

- **NULL is NOT a value** - is a representation of the fact that there is NO VALUE
- Reasons for a NULL:
 - **VALUE NOT APPLICABLE** -
 - EMP relation - empno, deptno, salary, **commission**
 - **commission only applies to staff in sales dept.**
 - **VALUE UNKNOWN** -
 - Joe's salary is NULL, **Joe's salary is currently unknown**
 - **VALUE DOES NOT EXIST** -
 - **Tax File Number** - is applicable to all employees BUT Joe may not have a number at this time
 - **VALUE UNDEFINED** -
 - Certain items explicitly undefined eg. divide by zero
 - Columns Number_of_payments, Total_payments
 - Column Average_payment_made
 - **If Number_of_payments = 0 => Average undefined**

Thank you.