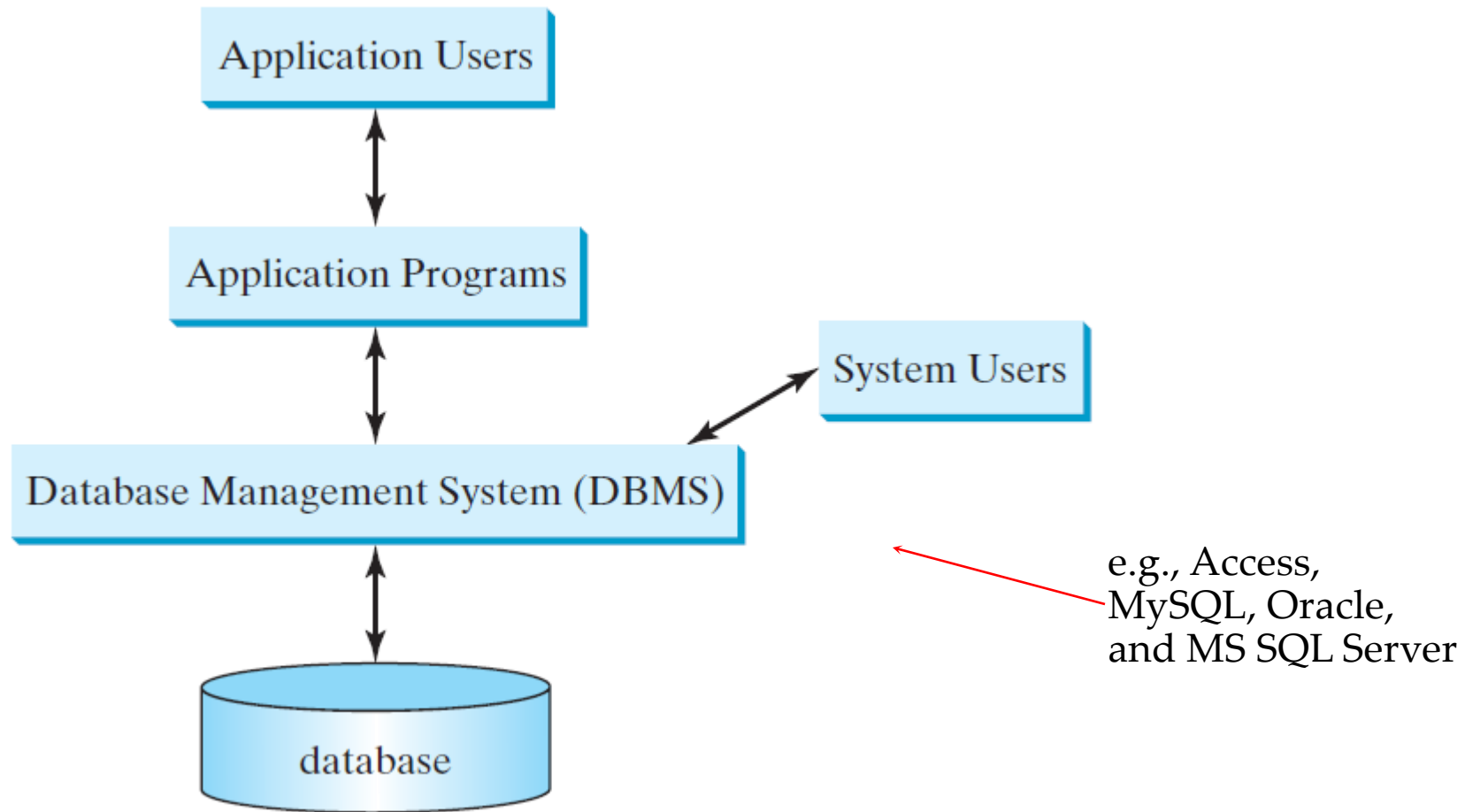


# Introduction to Java for C++ Programmers

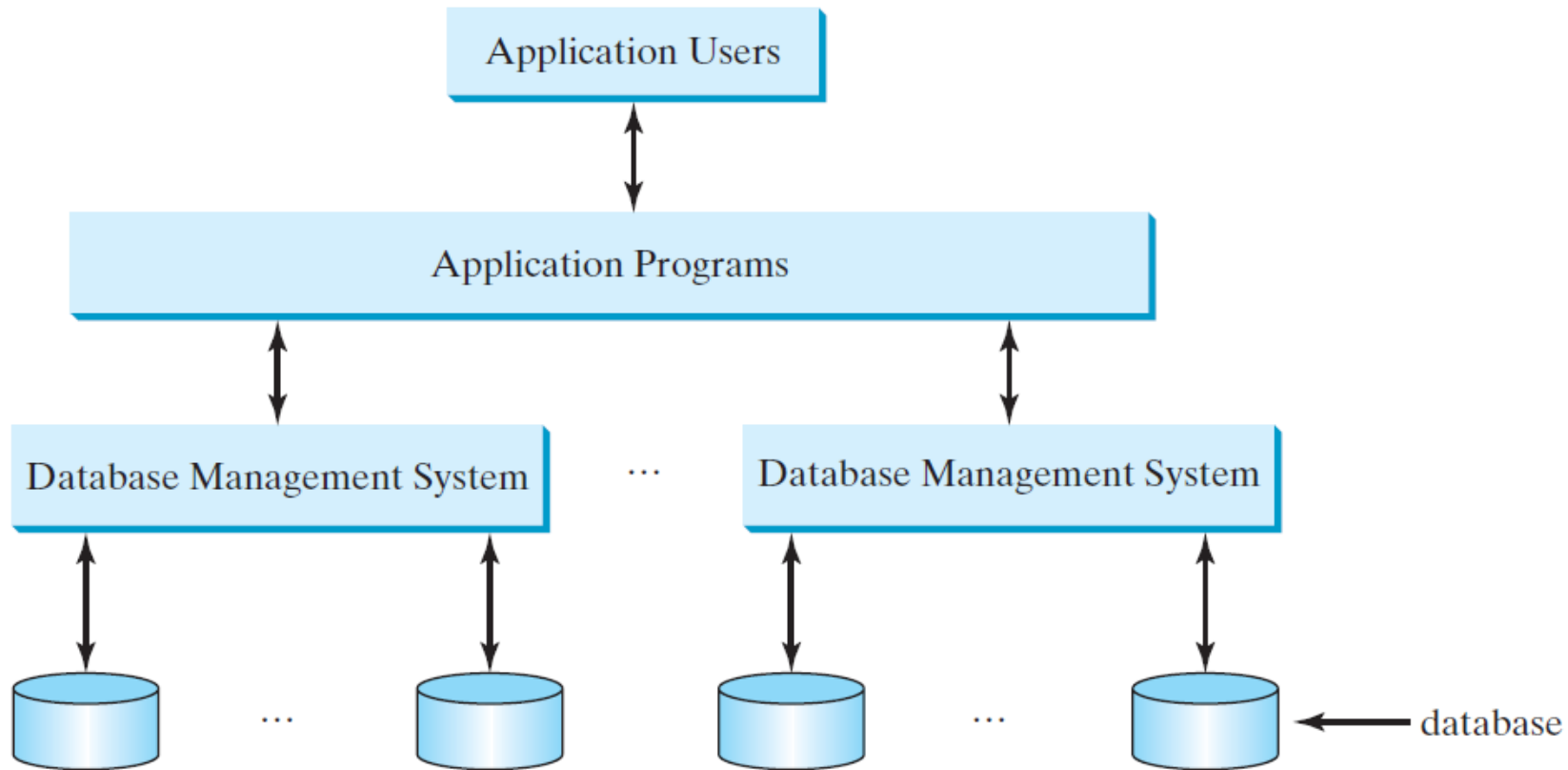
Introduction to Database

By: Mahboob Ali

# What is a Database System?



# Database Application Systems



# Rational Database and Relational Data Model

- Most of today's database systems are relational database systems, based on the relational data model. A relational data model has three key A relational data model has three key components
  - *Structure* defines the representation of the data.
  - *Integrity* imposes constraints on the data.
  - *Language* provides the means for accessing and manipulating data.

# Relational Structure

- A relational database consists of a set of relations.
- A relation has two things in one:
  - a *schema* (*which is a relation*)
  - an *instance* of the schema (which nothing but a table with rows and columns).

# Course Table

Relation/Table Name

↓

Course Table

Tuples/  
Rows →

Columns/Attributes

courseId	subjectId	courseNumber	title	numOfCredits
11111	CSCI	1301	Introduction to Java I	4
11112	CSCI	1302	Introduction to Java II	3
11113	CSCI	3720	Database Systems	3
11114	CSCI	4750	Rapid Java Application	3
11115	MATH	2750	Calculus I	5
11116	MATH	3750	Calculus II	5
11117	EDUC	1111	Reading	3
11118	ITEC	1344	Database Administration	3

# Student Table

Student Table									
ssn	firstName	mi	lastName	phone	birthDate		street	zipCode	deptID
444111110	Jacob	R	Smith	9129219434	1985-04-09	99	Kingston Street	31435	BIOL
444111111	John	K	Stevenson	9129219434	null	100	Main Street	31411	BIOL
444111112	George	K	Smith	9129213454	1974-10-10	1200	Abercorn St.	31419	CS
444111113	Frank	E	Jones	9125919434	1970-09-09	100	Main Street	31411	BIOL
444111114	Jean	K	Smith	9129219434	1970-02-09	100	Main Street	31411	CHEM
444111115	Josh	R	Woo	7075989434	1970-02-09	555	Franklin St.	31411	CHEM
444111116	Josh	R	Smith	9129219434	1973-02-09	100	Main Street	31411	BIOL
444111117	Joy	P	Kennedy	9129229434	1974-03-19	103	Bay Street	31412	CS
444111118	Toni	R	Peterson	9129229434	1964-04-29	103	Bay Street	31412	MATH
444111119	Patrick	R	Stoneman	9129229434	1969-04-29	101	Washington St.	31435	MATH
444111120	Rick	R	Carter	9125919434	1986-04-09	19	West Ford St.	31411	BIOL

# Enrollment Table

Enrollment Table			
ssn	courseId	dateRegistered	grade
444111110	11111	2004-03-19	A
444111110	11112	2004-03-19	B
444111110	11113	2004-03-19	C
444111111	11111	2004-03-19	D
444111111	11112	2004-03-19	F
444111111	11113	2004-03-19	A
444111112	11114	2004-03-19	B
444111112	11115	2004-03-19	C
444111112	11116	2004-03-19	D
444111113	11111	2004-03-19	A
444111113	11113	2004-03-19	A
444111114	11115	2004-03-19	B
444111115	11115	2004-03-19	F
444111115	11116	2004-03-19	F
444111116	11111	2004-03-19	D
444111117	11111	2004-03-19	D
444111118	11111	2004-03-19	A
444111118	11112	2004-03-19	D
444111118	11113	2004-03-19	B



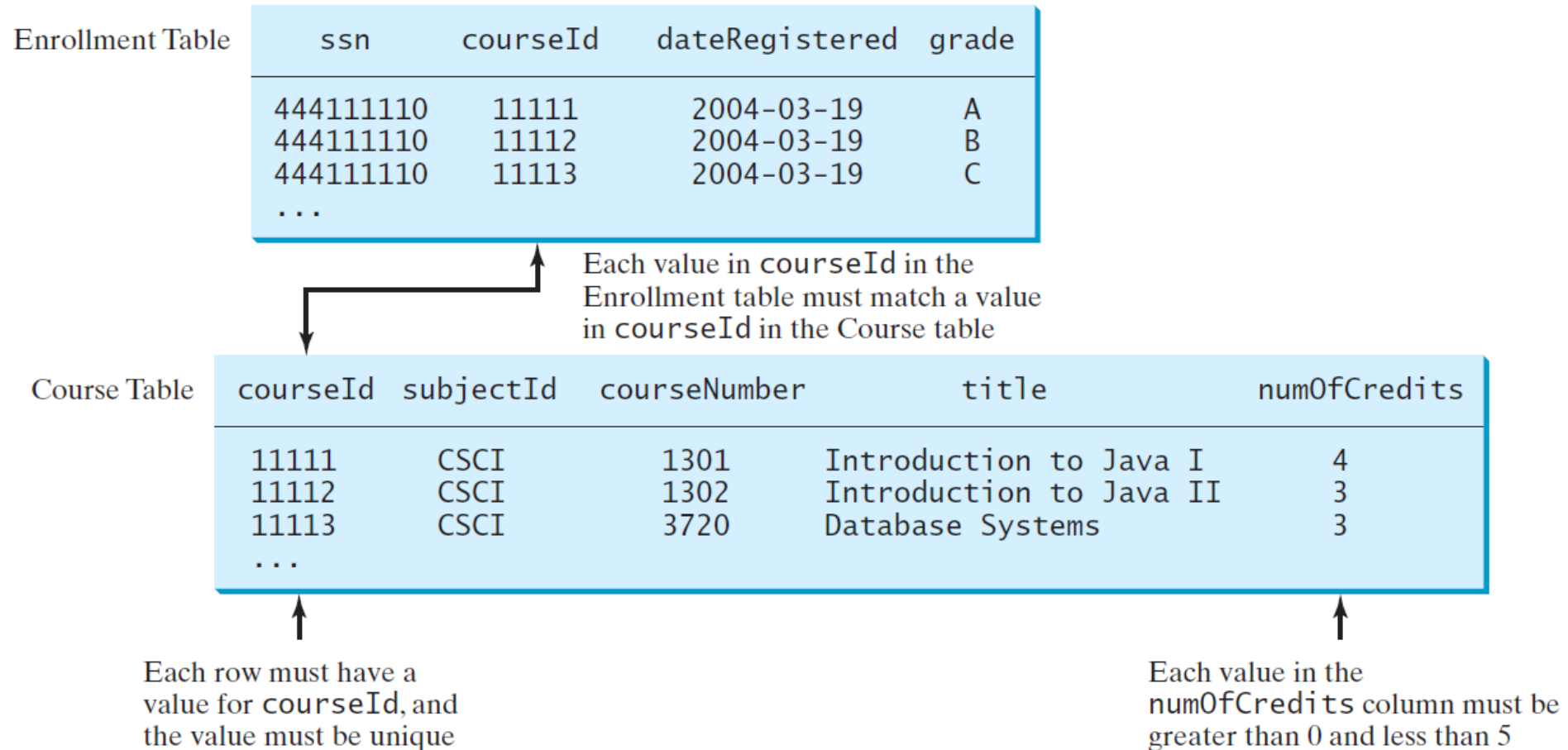
# Table vs. File

- A table or a relation is not same as a file.
- Most of the relational database systems store multiple tables in a file.

# Integrity Constraints

- An integrity constraint imposes a condition that all legal instances of the relations must satisfy.
- In general, there are three types of constraints:
  - *domain constraint*,
  - *primary key constraint*, and
  - *foreign key constraint*.
- Domain constraints and primary key constraints are known as *intra-relational constraints*, meaning that a constraint involves only one relation.
- The foreign key constraint is known as *inter-relational*, meaning that a constraint involves more than one relation.

# Domain Constraints



# Primary Key Constraints

Enrollment Table

ssn	courseId	dateRegistered	grade
444111110	11111	2004-03-19	A
444111110	11112	2004-03-19	B
444111110	11113	2004-03-19	C
...			

Each value in courseId in the Enrollment table must match a value in courseId in the Course table

Course Table

courseId	subjectId	courseNumber	title	numOfCredits
11111	CSCI	1301	Introduction to Java I	4
11112	CSCI	1302	Introduction to Java II	3
11113	CSCI	3720	Database Systems	3
...				

Each row must have a value for couserId, and the value must be unique

*Primary  
key  
constraint*

Each value in the numOfCredits column must be greater than 0 and less than 5

# Foreign Key Constraints

Enrollment Table

ssn	courseId	dateRegistered	grade
444111110	11111	2004-03-19	A
444111110	11112	2004-03-19	B
444111110	11113	2004-03-19	C
...			

Course Table

courseId	subjectId	courseNumber	title	numOfCredits
11111	CSCI	1301	Introduction to Java I	4
11112	CSCI	1302	Introduction to Java II	3
11113	CSCI	3720	Database Systems	3
...				

Each value in courseId in the Enrollment table must match a value in courseId in the Course table

Each row must have a value for courseId, and the value must be unique

*Foreign key constraint*

Each value in the numOfCredits column must be greater than 0 and less than 5

# Domain Constraints

- *Domain constraints* specify the permissible values for an attribute.
- Domains can be specified using standard data types such as integers, floating-point numbers, fixed-length strings, and variant-length strings.

# Domain Constraints Example

```
create table Course (  
    courseId char(5),  
    subjectId char(4) not null,  
    courseNumber integer,  
    title varchar(50) not null,  
    numOfCredits integer,  
    constraint greaterThanOne  
        check (numOfCredits >= 1) );
```

# Primary Key

- The *primary* key is one of the candidate keys designated by the database designer. The primary key is often used to identify tuples in a relation.

```
create table Course (  
    courseId char(5),  
    subjectId char(4) not null,  
    courseNumber integer,  
    title varchar(50) not null,  
    numOfCredits integer,  
    primary key (courseId)  
);
```



# Primary Key Constraints

- The *primary key constraint* specifies that the primary key value of a tuple cannot be null
- No two tuples in the relation can have the same value on the primary key.

# Foreign Key Constraints Formal Definition

- Formally, a set of attributes  $FK$  is a *foreign key* in a relation  $R$  that references relation  $T$  if it satisfies the following two rules:
- The attributes in  $FK$  have the same domain as the primary key in  $T$ .
- A non-null value on  $FK$  in  $R$  must match a primary key value in  $T$ .



# Foreign Key Example

```
create table Enrollment (  
    ssn char(9) ,  
    courseId char(5) ,  
    dateRegistered date ,  
    grade char(1) ,  
    primary key (ssn, courseId) ,  
    foreign key (ssn) references Student ,  
    foreign key (courseId) references Course  
);
```

# Foreign Key Discussion

- A foreign key is not necessarily the primary key or part of the primary in the relation.
- For example, subjectCode is a foreign key in the Course table that references the Subject table, but it is not the primary key in Course.

# SQL

- Structured Query Language, pronounced S-Q-L, or Sequel
- To access or write applications for database systems, you need to use the Structured Query Language (SQL). SQL is the universal language for accessing relational database systems. Application programs may allow users to access database without directly using SQL, but these applications themselves must use SQL to access the database.

•

# Examples of simple SQL statements

Create table

Drop table

Describe table

Select

Insert

Delete

Update

```
create table Course (  
  courseId char(5),  
  subjectId char(4) not null,  
  courseNumber integer,  
  title varchar(50) not null,  
  numOfCredits integer,  
  primary key (courseId)  
);
```

```
create table Student (  
  ssn char(9),  
  firstName varchar(25),  
  mi char(1),  
  lastName varchar(25),  
  birthDate date,  
  street varchar(25),  
  phone char(11),  
  zipCode char(5),  
  deptId char(4),  
  primary key (ssn)  
);
```

# Examples of simple SQL statements

Create table

Drop table

Describe table

Select

Insert

Delete

Update

```
drop table Enrollment;
```

```
drop table Course;
```

```
drop table Student;
```

# Examples of simple SQL statements

Create table

Drop table

Describe table

Select

Insert

Delete

Update

```
describe Course; -- Oracle
```



# Examples of simple SQL statements

Create table

Drop table

Describe table

Select

Insert

Delete

Update

```
select firstName, mi, lastName  
from Student  
where deptId = 'CS';
```

```
select firstName, mi, lastName  
from Student  
where deptId = 'CS' and zipCode = '31411';
```

```
select *  
from Student  
where deptId = 'CS' and zipCode = '31411';
```

# Examples of simple SQL statements

Create table

Drop table

Describe table

Select

Insert

Delete

Update

```
insert into Course (courseId, subjectId, courseNumber,  
title)  
values ('11113', 'CSCI', '3720', 'Database Systems', 3);
```

# Examples of simple SQL statements

Create table

Drop table

Describe table

Select

Insert

Update

Delete

```
update Course  
set numOfCredits = 4  
where title = 'Database Systems';
```

# Examples of simple SQL statements

Create table

Drop table

Describe table

Select

Insert

Update

Delete

```
delete Course  
where title = 'Database System';
```