



---

# DATABADSE SHORT NOTE

---



**Ketabton.com**

SEPTEMBER 21, 2019

# بسم الله الرحمن الرحيم

## Database

Database is an organized collection of logically related data

د ډیټابیس اساس Table دي

ډیټا بیس دوه اصلي وظیفې لري

- **Record keeping** د معلوماتو ساتل
- **Retrieval** د معلوماتو د موندلو او بیرته راكولو وړتیا

Data: ډیټا هغه خام مواد دی چې پروسیس پري نه وی اجرا شوی

Information: اجرا شوی ، منظم شوي او ترتیب شوی ډیټا ته ویل کیږي.

□ Examples of Information: د انفرمیشن ډیټا مثالونه

- List of students studying DBA
- Exam grades of a particular student
- Bank statement of the past 3 months
- **Data vs Information** د ډیټا او انفرمیشن تر منځ توپيرونه
  - د انفرمیشن لپاره ډیټا شرط ده
  - انفرمیشن output مواد دي
  - انفرمیشن product یا محصول مواد دي

• ډیټا لپاره انفرمیشن شرط نه دی

• ډیټا input مواد دي

• ډیټا خام مواد دي

**Database استعمال ځایونه**

Banking: all transactions ✓

- Airlines: reservations, schedules ✓
- Universities: registration, grades ✓
- Sales: customers, products, purchases ✓
- Manufacturing: production, inventory, orders, supply chain ✓
- Human resources: employee records, salaries, tax ✓  
deductions
- And many more... ✓

له ډیټا بیس څخه مخکی څه شی استعمالیدل؟

✓ کاغذی سیستم: مخکی ډیټا په کاغذونو کې لیکل کیدل او په هارډفایل کې  
نخیره کیدل

تاوانونه:

- د Security نشتوالی
- Recode پیدا کول پکې مشکل وو
- وخت ضایع کیدل
- په ډیر مشکل سره پکې Edit کیدل
- ✓ Flat file system: په دغه سیستم کې ډیټا په هادي Document کې  
نخیره کیدله لکه, Excel, word, .....

**Limitations of Flat File Systems:** دغه ډول ډیټابیس ځینې ستونزې درلودل

- Separation and isolation of data
- Duplication of data
- Data dependence
- Incompatible file formats

## Database Systems

د ډیټابیس سیستم هغه برخه چې ډیټابیس بلل کیږي هغه د ټیبلونو یوه مجموعه ده  
ډیټا بیس سیستم له څلورو شیانو څخه جوړ شوی چې په لاندې ډول دي

The database

The DBMS

Application Programs

Users

### Database Management System (DBMS)

د (DBMS) پواسطه کولای شو چي ډیټابیس create یا جوړ کړو

(DBMS) وظیفه Management and control access دی

(DBMS) یوزر ته لاندې سهولتونه رامنځته کولای شي

- Define a database د ډیټابیس مشخصول
- Create a database د ډیټابیس جوړول
- Maintain a database ستونزي د ډیټابیس پواسطه له منځه وړل
- Control access to the database.

څوک کولای شي چي ډیټابیس ته لاس رسی ورته پیدا کړي

(DBMS) اصلي هدف دادی چي ډیټابیس ته معلوماتو ستور کول او بیرته چي هر وخت وغواړو بیرته لاس ته راوړل په اسانۍ او موثر ډول سره.

### :Facilities provided by a DBMS

DBMS کومي اسانتیاوي رامنځته کوي:

#### Data Management . 1

:DDL (Data Definition Language)

په DDL کی مور ډیټابیس مشخصو ، رامنځته کوو یی او ډیټا ورداخلوو

DML (Data Manipulation Language)

په DML کی یوزر کولای شي چي ډیټا retrieve او insert, update, delete, کولو

په **DML** برخه کې Query language څخه استفاده کوو لکه SQL

## Controlled access: . 2

دا د هر چا د لاس رسي څخه مخنیوی کوي مثالونه یی په لاندې ډول دي

**security system**: دیتابیس د هر چا له لاس رسي څخه مخنیوی کوي

**integrity system** باید دیتابیس په هر حالت کې ثابت وي نه دا چې په یو ځای کې یو ډول او بل ځای کې بل ډول وي

**concurrency control system**: په عینې وخت کې زیات یوزران یو دیتابیس ته داخلیدل او سیستم ټولو ته شیر کول

**recovery control system**: د دیتابیس له منځه تللو په وخت کې د recovery سیستم درلودل

## DBMS Examples

- Popular DBMS products are
  - Microsoft Access
  - Microsoft SQL Server
  - Oracle Corporation's MySQL
  - Oracle Corporation's Oracle Database
  - IBM's DB2

## Database Application

دیتابیس اپلیکیشن د یوزر او DBMS ترمنځ ارتباط جوړوي او وظیفه یی **SQL statements** چې مور لیکو هغه DBMS ته انتقالوي

اپلیکیشن پروگرامونه کولای شو د سوفتویر جوړونکي لخوا تر لاسه کړو

مثال: که چیرې اکسس پروگرام تر لاسه کوو باید له مایکروسافت څخه یی واخلو

## Users

هغه څوک چې دیتابیس استعمالوي او کار ورڅخه اخلي

## Roles in the Database Environment

څلور ډوله خلک دي چي په DBMS کار کوي په ورسره اړیکه نيسي

- |                                  |                    |
|----------------------------------|--------------------|
| Data and database administrators | 1. ډيټابيس موديران |
| Database designers               | 2. ډيزاينران       |
| Application developers           | 3. اپليکشن جوړونکي |
| End-users                        | 4. استعمالونکي.    |

### Data Administrators(DA)

ډيټابيس پلان او د جوړولو طريقه پکي موجوده ده په دی د ډيولوپر هيڅ کار نشته او ډيولوپر ته وربندول کيږي چي کوم ډول ډيټابيس جوړ کړل شي.

**Data Administrator:** لاندې وظيفي لري

- پلان جوړوي
- د مشکيلاتو لپاره معياري حل لاره پيدا کوي
- جوړونکي ته د ډيټابيس مشخصات ورپه گوته کوي

### Database Administrators(DBA)

دا مسوليت لري (DA) کوم پلان ورکړي وو هغه مطابق فزيکيلي ډول ډيټابيس جوړ کړي لاندې وظيفي لري

- د ډيټابيس جوړول او د هغه تطبيق
- ډيټابيس ته به څوک داخلیدلای شي او کوم ډول کارونه ترسره کولای شي
- د جوړ شوي سيستم ستونزي له منځه وړل
- سيستم بايد داسي عيار او جوړ کړي چي يوزر يی په اسانۍ سره استعمالولای شي

**DBA** تخنيکي کارونه ترسره کوي او **DA** د پوهي په واسطه پلان جوړوي

### Database Designers

مور دوه ډوله ډيزاين کولای شو

1. Logical database designers
2. Physical database designers.

### 1 -Logical database designers

### لاندې مشخصات لري

- ډیټا او د ډیټا صفات رامنځته رامنځه کوی
- د ډیټا تر منځ اړیکه رامنځه ته کول
- هغه شرایط چی په ډیټا لپاره وي باید په ډیټابیس کی ستور شي

### Logical database design

په عملي ډول د ډیټابیس رامنځته کول دی ته لوجیکل ډیټابیس ډیزاین ویل کیږي

### Database Designers

مور کوم ډول ډیټابیس ته طرحه جوړه کړي هغه ته د فزیکي شکل جوړولو لپاره DBMS ته ضرورت ده مور نشو کولای چي په ورد یا اکسل کي ډیټابیس رامنځته کړو

لاندی شیان پکي شامل دي

- مور ټیپلي رامنځته کوو او ټیپلي یو بل سره وصلو لپاره PRAMARY KEY لگوو
- او د ډیټابیس لپاره لازم سکيورتي ولگول شي

### A design process suggestion for Microsoft Access

1. Determine the purpose of the database.
2. Find and organize the information required.
3. Divide the information into tables.
4. Turn information items into columns.
5. Specify primary keys.
6. Set up the table relationships.
7. Refine the design.
8. Apply the normalization rules

1. د ډیټابیس هدف باید معلوم وي
2. منظم معلوماتو پیدا کول
3. معلومات په ټیپلونو باندی ویشل
4. د هر د ډول معلوماتو لپاره جلا جلا کالمونه جوړول
5. primary keys باید په څه شي باندی ولگوو



6. د ټيبلونو تر منځ ارتباط ورکول
7. ډيزاين باندې تجديد نظر کول
8. نورمال قوانين بايد پرې APPLY کړو

## Application Developers

کله چې ډيټابيس جوړ شي نو بايد اپليکيشن پروگرام ورته جوړ شي تر څو په اسانې سره کار ترسره کړي دغه مسوليت ددې خلکو دی

بايد **Developers** د retrieving data, inserting, updating, and deleting data اسانتيا رامنځته کړي

او دا پروگرامونه بايد 3<sup>RD</sup> او يا په 4<sup>TH</sup> کې جوړيږي

## End-Users

دا هغه يوزران دي چې د خپل ضرورت په اساس له ډيټابيس څخه گټه اخلي او زده کوی يی او **End-Users** بيه دوه ډوله دي

### 1. Naïve users

- دا له ډيټابيس سره هيڅ اشنایي نه لري
- دوی يواځې ځان دومره پوهه وي چې څرنگه کار تری اخستل کيږي

### 2. Sophisticated users

- هغه يوزران دي چې د ډيټابيس په هر څه باندې پوهيږي
- د ډيټابيس په اسانتيا باندې هم پوهيږي
- دا USER په high-level query language باندې هم پوهيږي او ځانته پروگرام هم جوړولای شي

## Database System Life Cycle

ډيټابيس د معلوماتي سيستم مهمه برخه ده. کله چې مور و غواړو ډيټابيس رامنځته کوو بايد ټولې اړتياوي په نظر کې ونيسو د ډيټابيس په رامنځته کولو کې لاندې شيان موجود دي



- Database Planning دیتابیس پلان کول
- Requirements collection and analysis دیتابیس اړتیاوي پوره کول
- Database Design د دیتابیس ډیزاین کول
- Application Design د اپلیکیشن ډیزاین
- Prototyping د دیتابیس یوه نمونه بنودل
- Implementation د دیتابیس عملی کول
- Data Conversion and Loading د ډیټا لوډل کول
- Testing تجربه کول
- And operational maintenance. او ستونزو له منځه وړل

**Database Planning:** ددی لپاره پلان جوړیږي چې مور به دیتابیس لپاره څه استعمالوو او د څه شي لپاره یی جوړوو او باید څنگه یی جوړ کړو او کوم هدف باید ولري

**Requirements Collection and Analysis:** هغه شرکت یا یو دوکان یا هري بلي برخی ته چې دیتابیس جوړوي یا د هغه په اړه ټول انفرمیشن ولري تر څو دیتابیس کومه غلطی ونه لری

او دغه معلومات د دیتابیس لپاره ضرورت دی

**Database Design:** دیتابیس درى مرحلي لری

- **Conceptual Database Design:** په دي کي يواځي د یوی برخی لپاره دیتابیس او د هغه ډول انتخابیږی

- **Logical Database Design:** د دیتابیس لپاره خیالي ډیزاین جوړیږي

- **Physical Database Design:** په دي کي یو فزیکي ډول دیتابیس رامنځته کوو د ډیټا لپاره شرط ټاکه Relationship پکي رامنځته کوو Security ورکوو...

**Application Design:** ددی لپاره چې له دیتابیس څخه په اسانی سره کار واخلو نو یو اپلیکیشن رامنځته کوو لکه په لاندی ډول

- Insert operation

- Update operation

- Delete operation

Print operation •

## User Interface Guidelines

- معنا دار سرلیک
- Field باید منظم وی
- کوم مخففیات چي کاروو هغه باید هر خای کي ثابت وي
- صحیح رنگ انتخابوول
- ډیټا باید د لیدو وړ وي
- د غلطی په وخت کي باید میسج راکړي
- د فیلډ لپاره تشریحاتي مسجونہ لیکل
- استعمال کي باید اسانه وی
- **Prototyping**: د یوزر لپاره یوه نمونه جوړول تر څو چي یوزر ته وښودل شي چی د ډیټابیس څنگه کار کوي
- This is an optional step in database system life cycle.

### Implementation

کله چی مور ډیټابیس جوړ کړو غواړو چی دا عملي کړوپه دي کي لاندې شیان موجود دی

- Implementation of DDL statements
- Implementation of GUI
- Implementation of user views.
- Implementation of application programs
- Implementation of DML statements
- Implementation of security and integrity controls.

### Testing

**Learnability**: یوزر په څومره وخت کي کولاي شي چی له سیستم سره بلد شي

**Performance:** څه ډول د يوزر لپاره کار ترسره کولای شي

**Robustness:** کله چې يوزر په سيستم کې غلطی رامنځته کوي نو سيستم څومره کولای شي چې د يوزر غلطی له منځه يوسي

**Recoverability:** که چيری له سيستم څخه معلومات ختم شو نو ايا سيستم کولای شي چې زمونږ معلومات Recovery کړي؟

## The Relational Model

نن سبا له په ډيټابيس کې له له ټولو نه زيات مهم موډل The Relational Model دی لومړی ځل په 1970م کال **Dr Edgar Frank Codd** لخوا رامنځته شو دا په IBM کمپنۍ کې کارکوونکی وو

په دې موډل کې دوه ډوله مفهوم شتون لري

1. ډيټابيس د ريکارډ او د شيانو د ساتلو لپاره استعمالیږي

2. ډيټابيس کې ډيټا په ټيبل کې ستور کيږي

په ډيټابيس کې thing ته **ENTITY** وايي او **table** ته **RELATION** وايي

**Relation:** دوه بعدي ټيبل ده چې له روو او کالم څخه جوړ شوي

او لاندي خصوصيات لري

- په Row کې Entity موجوده وي
- په کالم کې د Entity لپاره Attribute موجود وي
- هر Cell بيل بيل value لري
- په هر کالم کې بايد يو ډول ډيټا موجوده وي
- هر کالم بايد واحد نوم ولري
- د کالمونه ترتيب هيڅ مهم ندي
- د Row ترتيب هم مهم ندي
- هيڅ اجازه نشته چې دوه Row يو ډول ډيټا ونلري

## Relational Database Model Terminology

په Relational Database کی لاندی اصطلاحات موجود دي

|                 |               |                  |
|-----------------|---------------|------------------|
| <b>Table</b>    | <b>Row</b>    | <b>Column</b>    |
| <b>File</b>     | <b>Record</b> | <b>Field</b>     |
| <b>Relation</b> | <b>Tuple</b>  | <b>Attribute</b> |

**Integrity Constraints:** زمور ډیټابیس کی چي معلومات ستور کوو باید د هر وخت لپاره ثابت وي ددی لپاره لاندی دري اصول شتون لري ددی دريو اصولو هدف دادی چي مور د خپل ډیټابیس integrity رامنځته کړو چي زمور ډیټابیس څومره گټور دی

**1. The Domain Integrity Constraint:** ددی مانا دادی چي په یوکالم کی هر ډول ډیټا موجوده وي باید نوره ډیټا یی هم همدغه data type ولري

| Student ID | First Name | Department | Phone      | Address   |
|------------|------------|------------|------------|-----------|
| 100        | Ahmad      | IS         | 0786778899 | Khost     |
| 101        | Shaheen    | IT         | 0700800900 | Khost     |
| 102        | Mohsin     | IT         | 0777885885 | Paktika   |
| 103        | Kamal      | IS         | 0765889944 | Khost     |
| 104        | Rahim      | IS         | 0799454545 | Logar     |
| 105        | Rashid     | IT         | 0789123456 | Nangarhar |

Domaine integrity

**The Entity Integrity Constraint**: زموږ دیتا هره Row باید ځانگړي ایډي باید لري تر څو له duplicate څخه مخنیوي وشي مثال:

| Student ID | First Name | Department | Phone      | Address   |
|------------|------------|------------|------------|-----------|
| 100        | Ahmad      | IS         | 0786778899 | Khost     |
| 101        | Shaheen    | IT         | 0700800900 | Khost     |
| 102        | Mohsin     | IT         | 0777885885 | Paktika   |
| 103        | Kamal      | IS         | 0765889944 | Khost     |
| 104        | Rahim      | IS         | 0799454545 | Logar     |
| 105        | Rashid     | IT         | 0789123456 | Nangarhar |

unique value

د Relational database اساسي قاعده هم دغه ده

**The Referential Integrity Constraint**: که په ټیبل کې دیتا یو بل سره ارتباط ولري

نو په primary key سره ارتباط ورکوي

نو د Master table ته له child table ته ارتباط ورکوي

### Relational Model Keys

Key: په یو ټیبل کې یو یا زیات کالمونه key بلل کېږي چې یو Row مشخص کوي

A key can be unique or non-unique.

Types of Keys

Surrogate Key

Composite Key

Candidate Key

Primary Key

## Foreign Key

**Surrogate Key**: یو unique ID ده چي د DBMS لخوا ورته مشخصیږي او په Auto ډول د سیستم لخوا ورته اضافه کیږي او هېڅ بدلون پکې نه راځي

| Field Name | Data Type  |                            |
|------------|------------|----------------------------|
| CustomerID | AutoNumber | Surrogate key for CUSTOMER |
|            |            |                            |
|            |            |                            |
|            |            |                            |
|            |            |                            |

Surrogate key

**Composite Key**: دا هغه key ده چي یو یا زیاتو کالمونو څخه پکې استفاده شوي وی

**Candidate and Primary Keys**: زموږ ډیټابیس کي ټولي ټیپلي candidate key

بلل کیږي له هغوي څخه یو انتخابیږي او primary key پرې لگوو او کوم چي پاته شو هغوي ته alternate ویل کیږي

**Foreign Key**: دا د primary key معکوس دی په child Table باندې لگېږي او دا کي تکرار کیدای هم شی له بهر څخه په ټیپل باندې لگېږي ځکه ورته foreign key ویل کیږي

## Functional Dependency

د دوو Attribute تر منځ ارتباط ته Functional dependency ویل کیږي

یا هغه حالت ته ویل کیږي چي په یو ټیپل کي یو Attribute د نورو Attribute یا کالمونو په اړه معلومات ورکوي

او یو کالم د نورو کالمونو څخه نمایندېدی کوي

| Object Color | Weight | Shape |
|--------------|--------|-------|
| Red          | 50g    | Ball  |
| Blue         | 50g    | Cube  |
| Yellow       | 70g    | Cube  |



په دي کی object color د نورو کالمونو لپاره **determinant** او دا نور کالمونه په دی کالم پوري دی Depends

## Why Functional Dependency?

### ولي یی مور استعمالوو

- په کالمونو کې د Candidate Key پیدا کولو لپاره استعمالوو
  - مهم او مانا لرونکی کالم د primary key لپاره انتخابوو
  - د Normalization لپاره استعمالیږي
  - تکرار څخه مخنیوي کوي
- Primary and Candidate Keys Revisited**: یو یا یو نه زیات کالمونه چې د ټولو کالمونو څخه نمایندګي کوي primary key بلل کیږي
- همدغه تعریف د candidate key لپاره هم دي
- که چېرته په یو ټیبل کې candidate key موجوده نه وي نو مور دوه یا درې کالمونه سره یوځای کوو تر څو unique شي او primary key پري ولګوو

## Normalization

دهغي پروسې څخه عبارت دي کوم چې په ډاټابیس کې په یو داسی جوړښت یا Structure سره Organize یعنی منظمه او ترتیب کوي تر څو د تکراري ډاټا، بی قاعدې Insertion، بی قاعدې Update او بی قاعدې Deletion څخه مخنیوي وکړي.

لومړی راځو په بی قاعدې والي Anomalies باندې خبرې کوو تر څو د نارمل کولو اهمیت په روښانه ډول باندې واضح کړو

**Anomalies in DBMS**: کله چې ډاټابیس منظم شوی نه وي له درې ډوله بی قاعدې والي یا Anomalies سره مخامخ کیږو چې له Insertion, Update and Deletion څخه عبارت دي

**NOTE**: تر څو د پورتنیو بی قاعدیوالي یا Anomalies ستونزو ته حل ووايو او یادي ستونزي له منځه یوسو تر څو زموږ ډاټا په کامله توګه نارمل وي د Normalization څخه په ګټې اخیستې کولای شو پورته یاد شوي مشکلات حل او فصل کړو

**:Why Normalization?**

1. Rows contain data about an entity
2. Columns contain data about attributes of the entity
3. Cells of the table hold a single value
4. All entries in a column are of the same kind
5. Each column has a unique name
6. The order of the columns is unimportant
7. The order of the rows is unimportant
8. No two rows may hold identical sets of data values



کله چې یو ټیبل پورتنی خواص ولري **Relation Model** بلل کيږي خو بیا هم ستونزه لري نو ددی لپاره مور بیا هم **Normalization** عملیه سرته سوي ترڅو زموږ ډیټابیس کی ستونزه رامنځته نشي مثالونه یی په سلاډونو کی واضح دي

### Modification Problem 1

- که چیرته مور په ټیبل کی **Normalization** ونلرو نو لاندی ستونزی رامنځته کيږي
- په ټیبل کی به یوه **Entity** بی ضرورته بار بار تکرار کيږي
  - که مور وغواړو دغه **Entity** اډیټ کړو نو هر څومره ځله چې تکرار وي هغومره ځله به اډیټ کوو
  - که چیرته مور وغواړو چې یوه **Entity** ډیلیټ کړو نو له هغه سره نوره **Entity** هم ختمیږي

### Relational Design Principles

- که چیرته مور وغواړو چې یو داسی ټیبل رامنځته کړو چې هیڅ ستونزه ونلري لاندی اصول باید مراعت کړو ترڅو یو **well-form** ولرو
- هغه **Relation** ته مور **Well formed** ویلای شو چې هر **Determinate** باید **Primary key** ولري
  - هغه **Relation** چې **Well formed** نه وي نو مور کولای شو چې په دوو او یا څو برخو یی وویشو ترڅو **Well formed** ته بدل شي
- پورتنی دوه اصول د **Normalization** زړه بلل کيږي
- نوټ:** **Well formed** هغه فورم ته ویل کيږي چې په بڼه طریقې سره ډیزاین شوي وي

### Normal Forms

یو ټیبل په هغه حالت کی په نارمل فارم کی دی چې **(BCNF)** حالت ته راوړسیري

- First normal form** ✓
  - Second normal form** ✓
  - Third normal form** ✓
  - Fourth normal form** ✓
  - Fifth normal form** ✓
  - Domain/Key normal form** ✓
- لومړنی دري یی ډیر مهم دي

**First normal form (1NF):** یو ټیبل ته هغه وخت **(1NF)** ویل کيږي چې :

1. د **Relation** چې کوم اته شرطونه دي هغه یی باید پوره کړي وي چې مخکی مو ویلي دي
2. یوه **primary key** ولري
3. باید معلومات تکراري نه وي

**Second Normal Form (2NF):** یو ټیبل ته هغه وخت **(2NF)** ویل کيږي چې :

1. باید په **(1NF)** کی وي
2. ټول هغه **nonkey attributes** باید د **Composite key** پواسطه **Determinate** شوي وي

**Third Normal Form (3NF):** یو ټیبل ته هغه وخت (3NF) ویل کیږي چې:

1. هغه په 2NF کې وي

2. هېڅ non key Attribute د بل non key Attribute په واسطه نه Determinate کیږي

**Boyce Codd Normal Form (BCNF):** کله چې **Dr. Edgar Frank Codd** ډیټابیس رامنځته

کړ نو ډیر وخت نه وو وتلي چې ده د نورمل فارم په باره معلومات ورکړ او وویل: یو ټیبل کیدای شي چې په دریم نارمل فارم کې هم غیري نورمال حالت امکان لري

په نتیجه کې **codd** او **R. Boyce** په گډه سره یو بل نورمال حالت رامنځته کړ، (BCNF) یی ونومول او هغه غیري نورمال حالت چې په دریم نورمال حالت کې هم له منځه نه وي تللي هغه پکې له منځه ځي

یو Relation ته مور هغه وخت BCNF ویلای شو چې هره **determinant** یی یو Candidate key وي

تاسو کولای شئ چې په یوه جمله کې Normalization تر پایه وپېژنئ

❖ **I swear to construct my tables so that all nonkey columns are dependent on the key, the whole key, and nothing but the key, so help me Codd!**

## The Normalization Process

د Normalization ترسره کولو لپاره دري مرحلې موجود دي که چیرته مور دغه دري مرحلې ترسره کړي نو مور یو ټیبل Normalize کړی

1. په Relation یا Table کې Candidate key مشخصول

2. په یو ټیبل کې functional dependencies مشخصول

3. که چیرته په یو ټیبل کې هېڅ Determinant Candidate key نه وي نو دغه ټیبل not well formed نه دي نو دغه حالت صحیح کولو لپاره لاندې مرحلې سرته رسو

(a) مور ټیبل دوو برخو ویشوو او په نوی ټیبل کې functional dependency مشخصوو

(b) او په نوی ټیبل کې په functional dependency باندې Primary key لگوو

(c) او دوهم ټیبل کې چې د determinant وظیفه سرته رسوي Foreign Key لگوو

(d) او وروسته د دواړو ټیبلونو تر منځ referential integrity رامنځته کوو

نوټ: د referential integrity څخه مراد Relationship دي

د Normalization ټیبل رامنځته کولو لپاره دریم حالت بار بار تکرارو تر څو رامنځته شي

پورتنی مرحلي په مثالونو کی تشریح کوو

### Sample DESCRIPTION Relation and Data

PRESCRIPTION (PrescriptionNumber, Date, Drug, Dosage, CustomerName, CustomerPhone, CustomerEmailAddress)

| PrescriptionNumber | Date       | Drug  | Dosage | CustomerName  | CustomerPhone | CustomerEmailAddress  |
|--------------------|------------|-------|--------|---------------|---------------|-----------------------|
| P10001             | 10/17/2017 | DrugA | 10mg   | Smith, Alvin  | 575-523-2233  | ASmith@somewhere.com  |
| P10003             | 10/17/2017 | DrugB | 35mg   | Rhodes, Jeff  | 575-645-3455  | JRhodes@somewhere.com |
| P10004             | 10/17/2017 | DrugA | 20mg   | Smith, Sarah  | 575-523-2233  | SSmith@somewhere.com  |
| P10007             | 10/18/2017 | DrugC | 20mg   | Frye, Michael | 575-645-4566  | MFrye@somewhere.com   |
| P10010             | 10/18/2017 | DrugB | 30mg   | Rhodes, Jeff  | 575-645-3455  | JRhodes@somewhere.com |

### Step 1 of the Normalization Process

- په پورتنی ټیبل کی مور candidate key مشخصوو
- PrescriptionNumber مور ته Date, Drug, and Dosage مشخصوي یا بی Determent کوي که چیرته نسخه یواځي د یو کس لپاره وي نو بیا په دی صورت کی مشخصوي او PrescriptionNumber زموږ Candidate key بلل کیري خو دا امکان نلري ځکه چي یو ډاکتر ته زیات ناروغان راځي
- که چیرته دغه ټول فیلډونه په یو ټیبل کی موجود وی نو یواځنی PrescriptionNumber دي چي ټول فیلډونه Determent کولای شي
- خو ستونزه دلته نه حل کیري

### Step 2 of the Normalization Process

- په دغه مرحله کی مور functional dependencies مشخصوو
- په دي ټیبل کی CustomerEmailAddress یو functional dependencies کیدای شي

CustomerEmailAddress  $\longrightarrow$  (CustomerName, CustomerPhone)

خو کله کله ځینی کسان یو شان ایمل ادرس لري نو ددي ستونزي ختمولو لپاره بلي مرحلي ته داخلیرو

### Step 3 of the Normalization Process

- په دي مرحله کی مور هغه Determent معلوموو چي Candidate key نشي کیدلای لکه CustomerEmailAddress
- په دي مرحله کی چي ستونزه له منځه یوسو ټیبل په دوو برخو ویشوو

| CustomerEmailAddress  | CustomerName  | CustomerPhone |
|-----------------------|---------------|---------------|
| ASmith@somewhere.com  | Smith, Alvin  | 575-523-2233  |
| JRhodes@somewhere.com | Rhodes, Jeff  | 575-645-3455  |
| MFrye@somewhere.com   | Frye, Michael | 575-645-4566  |
| SSmith@somewhere.com  | Smith, Sarah  | 575-523-2233  |

| PrescriptionNumber | Date       | Drug  | Dosage | CustomerEmailAddress  |
|--------------------|------------|-------|--------|-----------------------|
| P10001             | 10/17/2017 | DrugA | 10mg   | ASmith@somewhere.com  |
| P10003             | 10/17/2017 | DrugB | 35mg   | JRhodes@somewhere.com |
| P10004             | 10/17/2017 | DrugA | 20mg   | SSmith@somewhere.com  |
| P10007             | 10/18/2017 | DrugC | 20mg   | MFrye@somewhere.com   |
| P10010             | 10/18/2017 | DrugB | 30mg   | JRhodes@somewhere.com |

اوس ٽيبل Normalize شو او ستونزه له منځه لاڙه

نور مثالونه يی په سلايدونو کي ڪتلاي شی

## The E-R Model (The Entity Relationship Model)

لومړی ځل لپاره په ۱۹۷۶ کال کي **Peter Chen** لخوا رامنځته شو او مهم عناصر يی دادي

**Entities** ▪

**Attributes** ▪

**Identifiers** ▪

**Relationships.** ▪

E-R Modal مور ته په ډيټابيس کي د Entity مجموعه رابښی او د سيستم عناصر تشریح کوي او هغه ډياگرام چي مور پکي E-R Modal ډيزاين کوو هغه ته Entity-Relationship Diagram

يا E-R Diagram وایی

**Entity**: دا يواځي يو شخص نه ده او نه يو ځاي يا يو ايجيکت ده هر هغه څه چي د هغي په اړه يوه اداره وغواړی چي معلومات وساتي Entity بلل کيږي

يا په بل ډول: په نړی کي چي هر څه دي د Entity په نامه ياديږي

او هره Entity خپل خصوصيات لري چي په لاندي مثال کي ليدلاي شی

EMPLOYEE, STUDENT, PATIENT **Person** ▪

STORE, UNIVERSITY, STATE, COUNTRY **Place** ▪

MACHINE, BUILDING, AUTOMOBILE **Object** ▪

SALE, REGISTRATION, RENEWAL **Event** ▪

ACCOUNT, COURSE, WORKCENTER **Concept** ▪

### Entity Class and Entity Instance

**Entity Class**: Entity ډگانو مجموعه چي يو ډول خواص ولري Entity Class بلل کيږي



Entity Class مور ته د Entity جوړښت هم په لاس راكوي چي دا كوم ډول Entity ده او هره Entity په E-R model كې يو نوم لري او Entity Class مور په غټو حروفو ليكو مثال  
EMPLOYEE

**Entity Instance**: كله چي مور د entity لپاره يو تيبل جوړ كړو نو دي تيبل كې هري entity ته كه لږ وي او كه زيات وي ټولو ته Entity Instance ويل كيږي

## Strong Entity vs Weak Entity Types

**Strong Entity Type**: هغه entity ده چي په نورو entity پوري تړاو ونه لري او ازاد او خپلواك وي مثال: Student, employee, automobile, course

**Weak Entity Type**: هغه entity ده چي د هغي موجوديت په نورو entity پسي تړاو ولري او ازاد او خپلواك نه وي Grade, salary, model, subject

او هغه entity چي په هغه پوري نوري week entity تړاو ولري نو هغه ته **identifier owner** يا **owner** ويل كيږي

## Attributes

هره entity يو يا څو attribute لري چي entity تشریح كوي

مثال EmployeeName, DateOfHire, and JobSkillCode.

**Properties of Attributes**: هر attribute ډيټاټايب لري

- Character ▪
- Numeric ▪
- Date ▪
- Currency, etc. ▪

## Types of Attributes

**Simple or Atomic Attribute**: دا هغه attribute دي چي په برخو نه وېشل كيږي

Examples: CAR (ID, Color, Brand) ▪

**Composite Attribute**: دا هغه attribute دي چي په برخو وېشل كيږي

Example: Name, Address

**Single Valued Attribute**: دا هغه attribute دي چي single ويلو په ټيبل کي ذخيره کوي

**Derived Attribute**: دا هغه attribute دي چي value يی محاسبه کيږي کله age, total price

**Identifier Attribute**: دا هغه attribute دي چي په ټول ټيبل کي unique وي

## Identifiers

هغه attribute چي د هغي پواسطه يوه entity مشخص کيدلی او پيژندل کيدلای شي

**Single Identifier**: هغه attribute چي په يواځي ځان له ټولي row څخه نمايندگي کولای شي  
single identifier بلل کيږي کله ID

**Composite Identifier**: دا هغه identifier ده چي د دوه يا څو attributes د يوځاي کولو څخه منځته راځي. مثال: (FirstName, LastName, PhoneExtension)

Identifier کيدای شي unique وي يا non-unique وي

**Unique Identifier**: هغه identifier دي چي يواځي له يو entity instance څخه نمايندگي کوي  
مثال: Employee Number

**Non-Unique Identifier**: هغه identifier دي چي له يو سيټ څخه نمايندگي کوي  
مثال: EmployeeName

## Identifiers vs Keys

Identifier له key سره ورته والي لري په relational model کي مگر دوه مهم توپيرونه لري

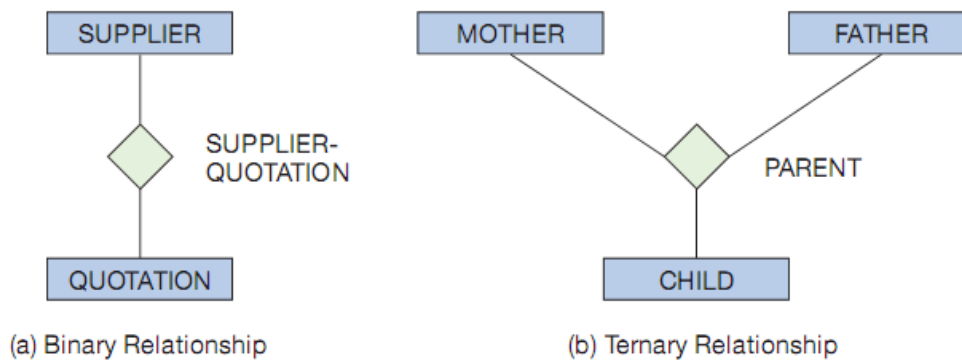
1. لومړي identifier فکري او خيالي مفهوم دي کيدای شي يو يا زيات attribute ولري
2. دوهم key به ضروري unique وي او Identifier کيدای شي unique وي يا non-unique وي

## Relationships

د دوو يا څو ټيبلونو لاجیکل ټراو ته Relationships وايي تر څو يو attributes له بل attributes سره ټراو ورکوي

**Binary and Ternary Relationships**: هغه Relationships چي له د دوو ټيبلونو تر منځ جوړ شوي وي هغه ته دويمه درجه يا binary relationships هم ويل کيږي

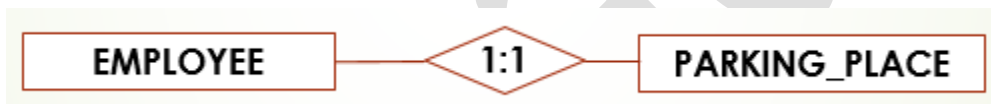
او که د دريو ټيبلونو تر منځ جوړ شوی وي هغه ته دريمه درجه يا ternary relationships هم ورته ويل کيږي



### Types of Binary Relationships

**The one-to-one (1:1) relationship**: هغه Relationships دي چي د يو ټيبل يواځي يو instance د بل ټيبل له يو instance سره تړاو لري

په اسانه ډول سره داسي وايو ( چي يو ټيبل له بل ټيبل سره يواځي په يو key سره ارتباط ولري one Relationships to one بلل کيږي )



**The one-to-many (1:N) relationship**: هغه Relationships دي چي د يو ټيبل يو instance د نورو ټيبلو له ډيرو instance سره ارتباط ولري

يا هغه Relationships ته ويل کيږي چي د يو ټيبل primary key د بل ټيبل د foreign key سره connect وي



**The many-to-many (N:M) relationship**: هغه Relationships دي چي د زياتو ټيبلونو instance له نورو زياتو ټيبلونو instance سره ارتباط لري



### Maximum and Minimum Cardinality

د Relationships دري ډوله چي مور يي مخکي يادونه وکړه دا د Cardinality پواسطه طبقه بندي شوي ده چي هدف يي شمير دي

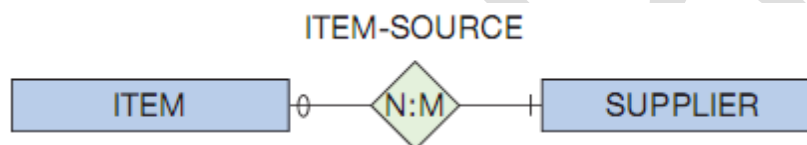


**Maximum Cardinality**: هغه Cardinality ته ويل کيږي چې د يو entity زيات instance په Relationships برخه اخستلاي شي



**Minimum Cardinality**: هغه Cardinality ته ويل کيږي چې د يو entity کم instance په Relationships برخه اخستلاي شي

او که چيرته Cardinality صفر وي نو ضروري نه ده چې خامخا دی په Relationships برخه واخلي



## SELECT & PROJECT Operations in Relational Algebra

**The SELECT Operation**: دا Operation ددي لپاره استعمالیږي چې څو فرعي ریکارډونه له يو ټیبل څخه انتخاب کړو

SELECT operation يو filter دي چې يواځي هغه ریکارډونه را اخلي کوم چې له condition سره برابر وي هغه condition چې مور ورته ورکړي وي او هغه ریکارډونه چې مور ته بنیي په افقي ډول وي

select operation دوه برخي لري

1. هغه ریکارډونه چې له condition سره برابر وي هغه مور بنیي
2. هغه ریکارډونه چې له condition سره برابر نه وي له هغو ریکارډونو څخه صرف نظر کيږي

select operation syntax په لاندې ډول ده

$\sigma_{\langle \text{selection condition} \rangle}(\mathbf{R})$  په دی کی سگما د select معنی ورکوي

لومړي سگما علامه بیا condition لیکو او ورسته د هغه ټیبل نوم لیکو له کوم ټیبل نه چې ریکارډونه را اخلو.

کیدای شي چې شرطونه له یو زیات وي مثال په ډول

$\sigma_{(\text{Dip}=4 \text{ AND Salary}>25000) \text{ OR } (\text{Dip}=5 \text{ AND Salary}>30000)}(\text{EMPLOYEE})$

مثالونه: د EMPLOYEE په نامه ټیبل څخه هغه ریکارډونه چې په 4 نمبر department کي کار کوي سلیکت کړي؟

$\sigma_{\text{Dept} = 4}(\text{EMPLOYEE})$

هغه کارکونگی راته انتخاب کړی چی معاش یی له 30000 څخه زیات وي  
 $\sigma_{\text{Salary} > 30000}(\text{EMPLOYEE})$

**The PROJECT Operation**: دا operation له ټیبل څخه مور ته کالمونه یا attribute انتخابوي او هغه کالمونه چي له condition سره برابر نه وي له هغه څخه صرف نظر کوی دا هم دوه برخي لري

1. هغه کالمونه راته انتخابوي چي له condition سره برابر وي
2. او له هغه کالمونه څخه صرف نظر کيږي چي له condition سره نه وي برابر وي

### Select operation Syntax

$\Pi_{\langle \text{projection condition} \rangle}(\text{R})$  په دغه کی د پای د PROJECT معنی لري

د هغو کارکونکو نومونه، تخلص او معاش راته انتخاب کړی

$\Pi_{\text{name, Lname, salary}}(\text{EMPLOYEE})$

مثال: د کارکونکو جنس او معاش راته انتخاب کړه  $\Pi_{\text{gender, salary}}(\text{EMPLOYEE})$

| Sex | Salary |
|-----|--------|
| M   | 30000  |
| M   | 40000  |
| F   | 25000  |
| F   | 43000  |
| M   | 38000  |
| M   | 25000  |
| M   | 55000  |

## The Set Operations in Relational Algebra

مور به دلته دري عمليي تشریح کوو چي عبارت دي له union, intersection, set difference (د سیتونو اتحاد، تقاطع او د سیتونو منفي عمليه) خو دلته د سیتونو څخه ټیبلونو مراد دي

او په دي U علامی سره بنوودل کيږي مثال: لومړي ټیبل U دوهم ټیبل

**UNION**: په دغه عمليه کي ټول هغه ریکارډونه چي په R او S ټیبلونو کي موجود وي يعني هغه ریکارډونه چي په R ټیبل کي موجود وي او په S په ټیبل کي هم موجود وي له دواړو څخه یو ریکارډ را اخیستل کيږي او په دریم ټیبل کي ځاي پرځاي کيږی او داسی ډول سره بنوودل کيږي  $R \cup S$

اتحاد په دي U علامی سره بنوودل کيږي مثال: لومړي ټیبل U دوهم ټیبل

د STUDENT او INSTRUCTOR دوه ټیبلونو اتحاد معلوم کړي

### STUDENT $\cup$ INSTRUCTOR

### Result(b) Table

(a) STUDENT

| Fn      | Ln      |
|---------|---------|
| Susan   | Yao     |
| Ramesh  | Shah    |
| Johnny  | Kohler  |
| Barbara | Jones   |
| Amy     | Ford    |
| Jimmy   | Wang    |
| Ernest  | Gilbert |

INSTRUCTOR

| Fname   | Lname   |
|---------|---------|
| John    | Smith   |
| Ricardo | Browne  |
| Susan   | Yao     |
| Francis | Johnson |
| Ramesh  | Shah    |

(b)

| Fn      | Ln      |
|---------|---------|
| Susan   | Yao     |
| Ramesh  | Shah    |
| Johnny  | Kohler  |
| Barbara | Jones   |
| Amy     | Ford    |
| Jimmy   | Wang    |
| Ernest  | Gilbert |
| John    | Smith   |
| Ricardo | Browne  |
| Francis | Johnson |

**INTERSECTION:** په دغه عملیه هغه ریکارډونه را اخیستل کيږي چې په دواړو ټیبلونو کې یوشان وي مثلاً یو ریکارډ په R ټیبل کې موجود ده او په S ټیبل کې هم موجود ده له دواړو ټیبلونو څخه همدغه یو ریکارډ را اخیستل کيږي او له نورو څخه صرف نظر کيږي،  $R \cap S$  تقاطع په دې علامې سره بنودل کيږي

د STUDENT او INSTRUCTOR دوه ټیبلونو تقاطع معلوم کړي

### STUDENT $\cap$ INSTRUCTOR

### Result © Table

(a) STUDENT

| Fn      | Ln      |
|---------|---------|
| Susan   | Yao     |
| Ramesh  | Shah    |
| Johnny  | Kohler  |
| Barbara | Jones   |
| Amy     | Ford    |
| Jimmy   | Wang    |
| Ernest  | Gilbert |

INSTRUCTOR

| Fname   | Lname   |
|---------|---------|
| John    | Smith   |
| Ricardo | Browne  |
| Susan   | Yao     |
| Francis | Johnson |
| Ramesh  | Shah    |

(c)

| Fn     | Ln   |
|--------|------|
| Susan  | Yao  |
| Ramesh | Shah |

**SET DIFFERENCE (or MINUS):** په دغه عملیه کې هغه ریکارډونه بنودل کيږي چې په

دواړو کې توپیر ولري مثال په R او S ټیبلونو کې دوه ریکارډونه یوشان دي او نور ټول مختلف دي

$R - S$  عملیه کې دغه دوه ریکارډونه ختمیږي نور ټول په دریم ټیبل کې بنودل کيږي

او د سیتونو تفریق په (-) علامې باندې بنودل کيږي مثال: لومړي - دوهم ټیبل

$R \cap S = S \cap R$  او  $R \cup S = S \cup R$  که لومړی هر ټیبل ولیکو هیڅ فرق نه کوي

مگر Set difference کې  $R - S \neq S - R$  نه دی سره مساوي

د STUDENT او INSTRUCTOR دوه ټيبلونو تفريق معلوم کړي

(a) STUDENT

| Fn      | Ln      |
|---------|---------|
| Susan   | Yao     |
| Ramesh  | Shah    |
| Johnny  | Kohler  |
| Barbara | Jones   |
| Amy     | Ford    |
| Jimmy   | Wang    |
| Ernest  | Gilbert |

INSTRUCTOR

| Fname   | Lname   |
|---------|---------|
| John    | Smith   |
| Ricardo | Browne  |
| Susan   | Yao     |
| Francis | Johnson |
| Ramesh  | Shah    |

(d)

| Fn      | Ln      |
|---------|---------|
| Johnny  | Kohler  |
| Barbara | Jones   |
| Amy     | Ford    |
| Jimmy   | Wang    |
| Ernest  | Gilbert |

STUDENT - INSTRUCTOR

Result (d) Table

INSTRUCTOR-SUDENT Result(e) Table

| Fname   | Lname   |
|---------|---------|
| John    | Smith   |
| Ricardo | Browne  |
| Francis | Johnson |

**The Cartesian Product:** دې ته CROSS PRODUCT يا CROSS JOIN هم ويل

کيږي چې د دوو ټيبلونو د ريکارډونو ضرب عمليه ده او (X) سره بنودل کيږي

دا هم binary set عمليه ده مگر هغه ټيبلونه چې دغه عمليه ورباندې سرته رسوو بايد د ټيبلونو عناصر مختلف وي او ددې عمليې محصول د ټولو ټيبلونو ريکارډونه په نوي ټيبل کې ځاي پر ځاي کيږي

که چيرته د دوو ټيبلونو attributes يو ډول نومونه لري مثال ID او يا name په دواړو ټيبلونو کې موجود دی نو دلته ضرب عمليه نه ترسره کيږي سيستم دا نه پيژني چې دا اوس د student name ده که د customer name ده

دا ستونزه د ختمولو لپاره لومړی د ټيبل نوم او بيا ډاټ لیکو او وروسته attribute لیکو مثال:

Rbooks x Rlendings (id, name, author,id,name,bookid)

دا ستونزه دا ډول حل کوو چې د Rename په نوم يادېږي (books.id, . . . , lendings.id, . . .)

**The JOIN Operation:** په دغه عملیه کې له دوو ټیبلونو څخه ډیټا یو ټیبل ته رازي مثال یو ټیبل کې دوه ریکارډونه دی او بل ټیبل کې دري ریکارډونه دي دواړه چې سره شي پینځه ریکارډونه کيږي دا عملیه په دغه  $\bowtie$  سره بنودل کيږي

که چیرته مور و غواړو چې له دوو ټیبلونو څخه معلومات را واخلو نو د **Join** عملیې سره په اسانۍ سره کولای شو تر څو کار ترسره کړو

## Type of Join Operation

1. **THETA JOIN:** په دغه کې ټول هغه ریکارډونه له دواړو ټیبلونو څخه را اخلې کوم چې مور

$$R_1 \bowtie_{A\theta B} R_2$$

condition ورته ورکړي وي مثال :

ټیټا  $\theta$  به له Relation operator  $\{<, \leq, =, >, \geq\}$  څخه یو وي

2. **EQUIJOIN:** په دغه عملیه کې هر وخت د مساوي علامه کارول شوي وي

$$R_1 \bowtie_{A=B} R_2 = \sigma_{A=B}(R_1 \times R_2)$$

3. **NATURAL JOIN:** دغه Join هغه وخت ترسره کيږي چې د attribute نومونه یوشان

وي په دغه حالت کې به د Rename عملیه ترسره کوي

مثال:  $R_1, R_2$  ټیبلونه څخه دا  $X_1, X_2, X_3$  join کوو

$$(R_1.X_1 = R_2.X_1) \wedge (R_1.X_2 = R_2.X_2) \wedge (R_1.X_3 = R_2.X_3)$$

د Join نور ډولونه هم شته (Semi Join, Outer Join (Left Outer Join, Right Outer Join, Full Outer Join))

## Summary of Relational Algebra Operations

| OPERATION    | PURPOSE  | NOTATION   |
|--------------|--|--|
| SELECT       | Selects all tuples that satisfy the selection condition from a relation $R$ .  | $\sigma_{\langle \text{selection condition} \rangle}(R)$   |
| PROJECT      | Produces a new relation with only some of the attributes of $R$ , and removes duplicate tuples.  | $\pi_{\langle \text{attribute list} \rangle}(R)$   |
| THETA JOIN   | Produces all combinations of tuples from $R_1$ and $R_2$ that satisfy the join condition.  | $R_1 \bowtie_{\langle \text{join condition} \rangle} R_2$  |
| EQUIJOIN     | Produces all the combinations of tuples from $R_1$ and $R_2$ that satisfy a join condition with only equality comparisons.   | $R_1 \bowtie_{\langle \text{join condition} \rangle} R_2$ , OR<br>$R_1 \bowtie_{\langle \text{join attributes 1} \rangle, \langle \text{join attributes 2} \rangle} R_2$       |
| NATURAL JOIN | Same as EQUIJOIN except that the join attributes of $R_2$ are not included in the resulting relation; if the join attributes have the same names, they do not have to be specified at all. | $R_1 *_{\langle \text{join condition} \rangle} R_2$ ,<br>OR $R_1 *_{\langle \text{join attributes 1} \rangle, \langle \text{join attributes 2} \rangle} R_2$<br>OR $R_1 * R_2$ |



|                   |  |                      |
|-------------------|--|----------------------|
| UNION             | Produces a relation that includes all the tuples in $R_1$ or $R_2$ or both $R_1$ and $R_2$ ; $R_1$ and $R_2$ must be union compatible.                             | $R_1 \cup R_2$       |
| INTERSECTION      | Produces a relation that includes all the tuples in both $R_1$ and $R_2$ ; $R_1$ and $R_2$ must be union compatible.   | $R_1 \cap R_2$       |
| DIFFERENCE        | Produces a relation that includes all the tuples in $R_1$ that are not in $R_2$ ; $R_1$ and $R_2$ must be union compatible.  | $R_1 - R_2$          |
| CARTESIAN PRODUCT | Produces a relation that has the attributes of $R_1$ and $R_2$ and includes as tuples all possible combinations of tuples from $R_1$ and $R_2$ .                   | $R_1 \times R_2$     |
| DIVISION          | Produces a relation $R(X)$ that includes all tuples $t[X]$ in $R_1(Z)$ that appear in $R_1$ in combination with every tuple from $R_2(Y)$ , where $Z = X \cup Y$ . | $R_1(Z) \div R_2(Y)$ |

منحنی از موبنی پوری د Fundamental of Database لنڈ لیکچرنوت پای

Wiqade



**Get more e-books from [www.ketabton.com](http://www.ketabton.com)  
Ketabton.com: The Digital Library**