DATABADSE SHORT NOTE

Ketabton.com

SEPTEMBER 21, 2019

(c) ketabton.com: The Digital Library

بسم الله الرحمن الرحيم

Database

Database is an organized collection of logically related data

د ډيتابيس اساس Table دي

ډيټا بيس دوه اصلي وظيفي لري

- **Record keeping** د معلوماتو ساتل
- **Retrieval** د معلوماتو د موندلو او بيرته راکولو ورتيا

Data: ډيټا هغه خام مواد دی چي پروسيس پري نه وی اجرا شوی

Information: اجرا شوی ، منظم شوي او ترتيب شوی ډيټا ته ويل کيږي.

❑ Examples of Information: د انفرميشن ډيټا مثالونه

- List of students studying DBA
- Exam grades of a particular student
- Bank statement of the past 3 months

- **Data vs Information** **د ډيټا او انفرميشن تر منځ توپيرونه**
  - د انفرميشن لپاره ډيټا شرط ده
  - انفرميشن output مواد دي
  - انفرميشن product يا محصول مواد دي
  - ________________________________
  - ډيټا لپاره انفرميشن شرط نه دی
  - ډيټا input مواد دي
  - ډيټآ خام مواد دي

Database استعمال ځايونه

✓ Banking: all transactions

Wiqade

(c) ketabton.com: The Digital Library

- ✓ 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 مخکې څه شی استعمالیدل؟

✓ **کاغذي سیستم:** مخکي ډیټا په کاغذونو کي لیکل کیدل او په هارډفایل کی ذخیره کیدل

تاوانونه:

- د Security نشتوالي
- Recode پیدا کول پکي مشکیل وو
- وخت ضایع کیدل
- په ډیر مشکیل سره پکي Edit کیدل

✓ Flat file system: په دغه سیستم کي ډیټا په هادي Document کی ذخیره کیدله لکه word, Excel, ...

Limitations of Flat File Systems: دغه ډول ډیتابیس ځیني ستونزي درلودل

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

Database Systems

د ډیټآبیس سیستم هغه برخه چي ډیتابیس بلل کیږي هغه د ټیبلونو یوه مجموعه ده

ډیټا بیس سیستم له څلورو شیانو څخه جوړ شوی چی په لاندي ډول دي

Prepared: ABDUL RAHIM WIQADE Email: Wiqade99@Gmail.com Phone: 0764008879

(c) ketabton.com: The Digital Library

The database

The DBMS

Application Programs

Users

Database Management System (DBMS)

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

(DBMS) وظیفه Management and control access دی

(DBMS) یوزر ته لاندي سهولتونه رامنځته کولای شي

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

ځوک کولای شي چي ډیتابیس ته لاس رسی ورته پیدا کړي

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

:Facilities provided by a DBMS

DBMS کومي اسانتیاوي رامنځنته کوي:

Data Management . 1

:DDL (Data Definition Language)

په **DDL** کی موږ ډیتابیس مشخصوو ، رامنځته کوو یی او ډیتا ورداخلوو

DML (Data Manipulation Language)

په **DML** کی یوزر کولای شي چی ډیتا insert, update, delete, اوretrieve کوو

(c) ketabton.com: The Digital Library

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

Controlled access:. 2

دا د هر چا د لاس رسي څخه مخنيوی کوي مثالونه يی په لاندي ډول دي

security system: ډيتابيس د هر چا له لاس رسي څخه مخنيوی کوي

integrity system بايد ډيتابيس په هر حالت کي ثابت وي نه دا چي په يو ځای کي يو ډول او بل ځاي کي بل ډول وي

concurrency control system: په عيني وخت کي زيات يوزران يو ډيتابيس ته داخليدل او سيستم ټولو ته شير کول

recovery control system: د ډيټا له منځه تللو په وخت کی د recoveryسيستم 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

ډيتابيس اپليکيشن د يوزر او DBMS ترمنځ ارتباط جوړوي او وظيفه يی **SQL statements** چي موږ ليکو هغه DBMS ته انتقالوي

اپليکيشن پروګرامونه کولای شو د سوفتوير جوړونکی لخوا تر لاسه کړو

مثال : که چيری اکسس پروګرام تر لاسه کوو بايد له مايکروسافټ څخه يی واخلو

Users

هغه څوک چی ډيتابيس استعمالوی او کار ورڅخه اخلي

Roles in the Database Environment

(c) ketabton.com: The Digital Library

څلور ډوله خلک دي چي په DBMS کار کوی یه ورسره اړیکه نیسي

1. ډیتابیس مودیران Data and database administrators
2. ډیزاینران Database designers
3. اپلیکشن جوړونکي Application developers
4. استمالونکي. End-users

Data Administrators(DA)

ډیتابیس پلان او د جوړولو طریقه پکي موجوده ده په دی د ډیولوپر هیڅ کار نشته او ډیولوپر ته ورښودل کیږی چي کوم ډول ډیتابیس جوړ کړل شي.

Data Administrator: لاندي وظیفي لری

- پلان جوړوی
- د مشکیلاتو لپاره معیاري حل لاره پیدا کوي
- جوړونکي ته د ډیتابیس مشخصات ورپه ګوته کوي

Database Administrators(DBA)

دا مسولیت لري(DA) کوم پلان ورکړی وو هغه مطابق فزیکیلي ډول ډیتابیس جوړ کړی لاندي وظیفي لري

- د ډیتابیس جوړول او د هغه تطبیق
- ډیتابیس ته به څوک داخلیدلای شی او کوم ډول کارونه ترسره کولای شي
- د جوړ شوي سیستم ستونزي له منځه وړل
- سیستم باید داسي عیار او جوړ کړي چي یوزر یی په اسانی سره استعمالولای شی

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

Database Designers

موږ دوه ډوله ډیزاین کولای شو

1. Logical database designers
2. Physical database designers.

1 -Logical database designers

Prepared: ABDUL RAHIM WIQADE Email: Wiqade99@Gmail.com Phone: 0764008879

(c) ketabton.com: The Digital Library

لاندي مشخصات لري

- ډیټا او د ډیټا صفات رامنځته رامنځه کوی
- د ډیټا تر منځ اړیکه رامنځّه 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

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

لاندی شیان پکي شامل دي

- موږ ټیبلي رامنځته کوو او ټیبلي یو بل سره وصلو لپاره PRAMARY KEY لګوو
- او د ډیټابیس لپاره لازم سکیورټي ولګول 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 ټیبلونو باندی ویشل
4. د هرډ ډول معلوماتو لپاره جلا جلا کالمونه جوړول
5. primary keys باید په څه شي باندي ولګوو

(c) ketabton.com: The Digital Library

6. د ټيبلونو تر منځ ارتباط ورکول
7. ډيزاين باندي تجديد نظر کوول
8. نورمال قوانين بايد پري APPLY کړو

Application Developers

کله چي ډيتابيس جوړ شي نو بايد اپليکيشن پروګرام ورته جوړ شي تر څو په اساني سره کار ترسره کړي دغه مسوليت ددي خلکو دی

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

او دا پروګرامونه بايد 3RD او يا په 4TH کي جوړيږي

End-Users

دا هغه يوزران دي چي د خپل ضرورت په اساس له ډيتابيس څخه ګټه اخلي او زده کوی يی

او **End-Users** بپه دوه دوله دي

1. **Naïve users**
   - دا له ډيتآبيس سره هيڅ اشنايی نه لري
   - دوی يواځي ځان دومره پوهه وي چی څرنګه کار تري اخسيتل کيږي

2. **Sophisticated users**
   - هغه يوزران دي چی د ډيتابيس په هر څه باندی پوهيږي
   - د ډيتابيس په اسانتيا باندي هم پوهيږي
   - دا USER په high-level query language باندي هم پوهيږی او ځانته پروګرام هم جوړولای شي

Database System Life Cycle

ډيتابيس د معلوماتي سيستم مهمه برخه ده.
کله چی موږ وغواړو ډيتابيس رامنځته کوو بايد ټولي اړتياوي په نظر کی ونيسو د ډيتابيس په رامنځته کوولو کي لاندي شيان موجود دي

(c) ketabton.com: The Digital Library

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

Database Planning: ددی لپاره پلان جوړیږي چی موږ به ډیټابیس لپاره څه استعمالوو او د څه شي لپاره یی جوړوو او باید څنګه یي جوړ کړو او کوم هدف باید ولري

Requirements Collection and Analysis: هغه شرکت یا یو دوکان یا هري بلي برخی ته چي ډیټابیس جوړوي یا د هغه په اړه ټول انفرمیشن ولري تر څوډیټابیس کومه غلطي ونه لری

او دغه معلومات د ډیټابیس لپاره ضرورت دی

Database Design: ډیټابیس دری مرحلي لری

- **Conceptual Database Design:** په دي کي یواځي د یوی برخی لپاره ډیټابیس او د هغه ډول انتخابیږی
- **Logical Database Design:** د ډیټابیس لپاره خیالي ډیزاین جوړیږي
- **Physical Database Design:** په دي کي یو فزیکي ډول ډیټابیس رامنځته کوو

د ډیټا لپاره شرط ټاکه Relationship پکي رامنځته کوو Security ورکوو...

Application Design: ددی لپاره چی له ډیټابیس څخه په اسانی سره کار واخلو نو یو اپلیکیشن رامنځته کوو لکه په لاندي ډول

- Insert operation
- Update operation
- Delete operation

(c) ketabton.com: The Digital Library

- Print operation

User Interface Guidelines

- معنا دار سرلیک
- Fieldباید منظم وی
- کوم مخففات چي کاروو هغه باید هر ځای کي ثابت وي
- صحیح رنګ انتخابول
- ډیټا باید د لیدو وړ وي
- د غلطی په وخت کي باید میسج راکړي
- د فیلډ لپاره تشریحاتي مسجونه لیکل
- استعمال کي باید اسانه وی
- **Prototyping:** د یوزر لپاره یوه نمونه جوړول تر څو چي یوزر ته وښودل شي چی د ډیتابیس څنګه کار کوي
- This is an optional step in database system life cycle.

Implementation

کله چی موږ ډیټابیس جوړ کړو غواړو چی دا عملي کړوپه دي کي لاندي 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 شی چی له سیستم سره بلد شي

(c) ketabton.com: The Digital Library

Performance: څه ډول د یوزر لپاره کار ترسره کولاي شي

Robustness: کله چي یوزر په سیستم کي غلطی رامنځته کوي نو سیستم څومره کولای شی چي د یوزر غلطي له منځه یوسي

Recoverability: که چیری له سیستم څخه معلومات ختم شو نو ایا سیستم کولای شی چی زمونږ معلومات Recoveryکړی؟

The Relational Model

نن سبا له په ډیتابیس کی له له تولو نه زیات مهم موډلThe Relational Model دی

لومړی ځل په 1970م کال **,Dr Edgar Frank Codd** لخوا رامنځته شو دا په IBM کمپنی کی کارکوونکی وو

په دي موډل کي دوه ډوله مفهوم شتون لري

1. ډیتابیس د ریکارډ او د شیانو د ساتلو لپاره استعمالیږي
2. ډیتآبیس کی ډیتآ په تیبل کی ستور کیږي

په ډیتابیس کی thing ته **ENTITIY** وایي او **table** ته**RELATION** وایي

Relation: دوه بعدي تیبل ده چي له روو او کالم څخه جوړ شوي

او لاندي خصوصیات لري

- په Row کي Entity موجوده وي
- په کالم کي د Entity لپاره Attribute موجود وی
- هر Cell بیل بیل value لري
- په هر کالم کی باید یو ډول ډیتا موجوده وي
- هر کالم باید واحد نوم ولری
- د کالمونه ترتیب هیڅ مهم ندي
- د Row ترتیب هم مهم ندي
- هیڅ اجازه نشته چي دوه Row یو ډول ډیتا ونلري

(c) ketabton.com: The Digital Library

Relational Database Model Terminology

په Relational Database کی لاندي اصطلاحات موجود دي

Table	**Row**	**Column**
File	**Record**	**Field**
Relation	**Tuple**	**Attribute**

Integrity Constraints: زموږ ډیتابیس کي چي معلومات ستور کوو باید د هر وخت لپاره ثابت وي ددي لپاره لاندي دري اصول شتون لري

ددی دریو اصولو هدف دادي چي موږ د خپل ډیتابیس integrity رامنځته کړو چي زموږ ډیتابیس خومره ګټور دی

1. **The Domain Integrity Constraint:** ددی مانا دادي چي په یوکالم کي هر ډول ډیټا موجوده وي باید نوره ډیټاً 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

(c) ketabton.com: The Digital Library

The Entity Integrity Constraint: زموږ ډیټا هره Rowباید ځانګړي 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 کي ډیټاآ یو بل سره ارتباط ولري نو په primary keyسره ارتباط ورکوو

نو د Master table ته له child table ته ارتباط ورکوو

Relational Model Keys

Key: په یو ټیبل کی یو یا زیات کالمونه key بلل کیږي چې یو Row مشخص کوي

A key can be unique or non-unique.

Types of Keys

Surrogate Key

Composite Key

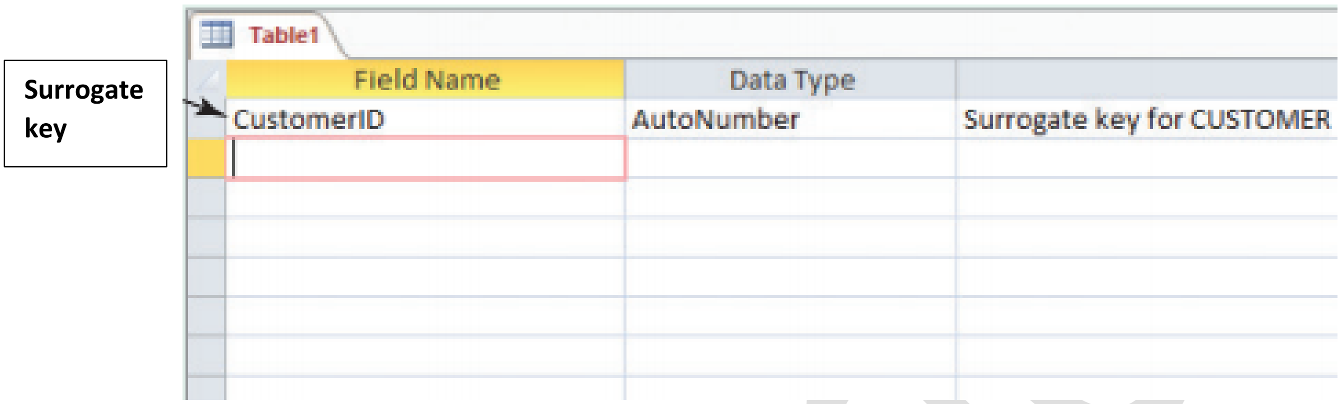
Candidate Key

Primary Key

(c) ketabton.com: The Digital Library

Foreign Key

Surrogate Key: يو unique ID ده چي د DBMS لخوا ورته مشخصيږي او په Auto ډول د سيستم لخوا ورته اضافه کيږي او هيڅ بدلون پکي نه راځي



Composite Key: دا هغه key ده چي يو يا زياتو کالمونو څخه پکي استفاده شوي وی

Candidate and Primary Keys: زمونږ ديتابيس کي ټولي تيبلي candidate key بلل کيږي له هغوي څخه يو انتخابيږي او primary key پري لګوو او کوم چي پاته شو هغوي ته alternate ويل کيږي

Foreign Key: دا د primary key معکوس دی په child Table باندي لګيږی او دا کي تکرار کيداي هم شی له بهر څخه په تيبل باندی لګيږي ځکه ورته foreign key ويل کيږي

Functional Dependency

د دوو Attribute تر منځ ارتباط ته Functional dependency ويل کيږی

يا هغه حالت ته ويل کيږي چي په يو تيبل کي يو Attribute د نورو Attribute يا کالمونو په اړه معلومات ورکوي

او يو کالم د نورو کالمونو څخه نمانيده ګی کوي

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

(c) ketabton.com: The Digital Library

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

Why Functional Dependency?

ولي يی موږ استعمالوو

- په کالمونو کي د Candidate Key پيدا کولو لپاره استمعالوو
- مهم او مانا لرروونکی کالم د primary key لپاره انتخابوو
- د Normalization لپاره استعماليږي
- تکرار څخه مخنيوي کوي

Primary and Candidate Keys Revisited: يو يا يو نه زيات کالمونه چي د ټولو کالمونو څخه نمايندګي کوي primary key بلل کيږي
همدغه تعريف د candidate key لپاره هم دي
که چيرته په يوتيبل کي candidate key موجوده نه وي نو موږ دوه يا دري کالمونه سره يوځای کوو تر څو unique شي او primary key پري ولګوو

Normalization

دهغي پروسي څخه عبارت دي کوم چي په ډاټابيس کی په يو داسی جوړښت يا Structure سره Organize يعني منظمه او ترتيب کوي تر څو د تکراري ډاټا، بي قاعدي Insertion، بي قاعدي Update او بي قاعدي Deletion څخه مخنيوي وکړي.
لومړی راځو په بي قاعدي والي Anomalies باندي خبري کوو تر څو د نارمل کولو اهميت په روښانه ډول باندي واضح کړو

Anomalies in DBMS: کله چي ډاټابيس منظم شوی نه وی له دري ډوله بي قاعدي والي يا Anomalies سره مخامخ کيږو چي له Insertion, Update and Deletion څخه عبارت دي

NOTE: تر څو د پورتنيو بي قاعديوالي يا Anomalies ستونزو ته حل ووايو او يادي ستونزي له منځه يوسو تر څو زمونږ ډاټا په کامله توګه نارمل وي د Normalization څخه په ګټي اخيستني کولاي شو پورته ياد 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

(c) ketabton.com: The Digital Library

کله چي يو ټيبل پورتني خواص ولري Relation Model بلل کيږي خو بيا هم ستونزه لري نو ددي لپاره موږ بيا هم Normalization عمليه سرته سوي ترڅو زموږ ډيټابيس کی ستونزه رامنځته نشي مثالونه يی په سلايډونو کي واضح دي

Modification Problem 1

که چيرته موږ په ټيبل کی Normalization ونلرو نو لاندي ستونزي رامنځته کيږي

- په ټيبل کی به يوه Entity بي ضرورته بار بار تکراريږي
- که موږ وغواړو دغه Entity اپډيټ کړو نو هر ځومره ځله چي تکرار وي هغومره ځله به آپډيټ کوو
- که چيرته موږ وغواړو چي يوه Entity ډيليټ کړو نو له هغه سره نوره Entity هم ختميږي

Relational Design Principles

که چيرته موږ وغواړو چي يو داسي ټيبل رامنځته کړو چي هيڅ ستونزه ونلري لاندي اصول بايد مراعت کړو تر څو يو well-form ولرو

- هغه Relation ته موږ Well formed ويلاي شو چي هر Determinate بايد Primary key ولري
- هغه Relation چي Well formed نه وي نو موږ کولاي شو چي په دوو او يا څو برخو يی وويشو تر څو Well formed ته بدل شي

پورتني دوه اصول د Normalization زړه بلل کيږي

نوټ: Well formed هغه فورم ته ويل کيږي چي په ښه طريقي سره ډيزاين شوي وي

Normal Forms

يو ټيبل په هغه حالت کي په نارمل فارم کي دی چي **(BCNF)** حالت ته راورسيږي

- ✓ **First normal form**
- ✓ **Second normal form**
- ✓ **Third normal form**
- ✓ **Fourth normal form**
- ✓ **Fifth normal form**
- ✓ **Domain/Key normal form**

لومړنی دري يی ډير مهم دي

(1NF) First normal form: يو ټيبل ته هغه وخت **(1NF)** ويل کيږي چي :

1. د Relation چي کوم اته شرطونه دي هغه يی بايد پوره کړي وي چي مخکي مو ويلي دي
2. يوه primary key ولري
3. بايد معلومات تکراري نه وي

Second Normal Form (2NF): يو ټيبل ته هغه وخت **(2NF)**ويل کيږي چي :

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

(c) ketabton.com: The Digital Library

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

1 . هغه په **2NF** کی وي

2 . هیڅ non key Attribute د بل non key Attribute پواسطه نه Determinate کیږي

Boyce Codd Normal Form (BCNF): کله چې **Dr. Edgar Frank Codd** ډیتابیس 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 کی **codd** او **R. Boyce** په ګډه سره یو بل نورمال حالت رامنځته کړ **(BCNF),** یې ونومول او هغه غیري نورمال حالت چې په دریم نورمال حالت کی هم له منځه نه وي تللي هغه پکي له منځه ځي

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

<u>تاسو کولای شی چې په یوه جمله کی Normalization تر پایه وپیژنی</u>

❖ <u>**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!**</u>

The Normalization Process

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

1 . په Relation یاTable کي Candidate key مشخصول

2 . په یو ټیبل کي functional dependencies مشخصول

3 . که چیرته په یو ټیبل کي هیڅ Determent،Candidate key نه وي نو دغه ټیبل not well formed نه دي نو دغه حالت صحیح کولو لپاره لاندی مرحلي سرته رسوو

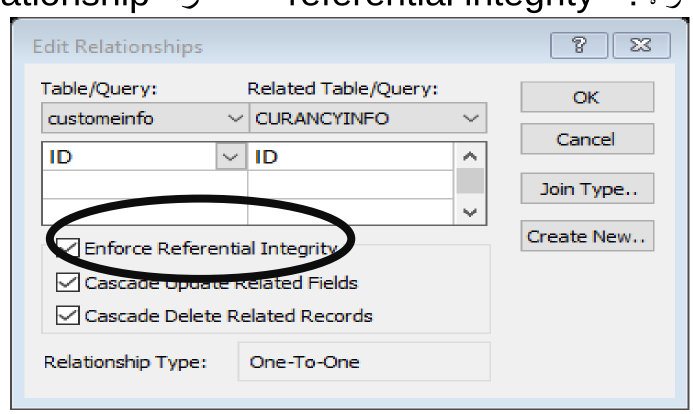
a) مونږ ټیبل دوو برخو ویشوو او په نوی ټیبل کی functional dependency مشخصوو

b) او په نوی ټیبل کي په functional dependency باندي Primary key لګوو

c) او دوهم ټیبل کي چې د determinant وظیفه سرته رسوي Foreign Key لګوو

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

نوټ: د referential integrity څخه مراد Relationship دي



(c) ketabton.com: The Digital Library

د Normalize تیبل رامنځته کولو لپاره دریم حالت بار بار تکراروو تر څو Normalization رامنځته شي

پورتني 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 پورتني تیبل کي موږ candidate key مشخصوو
- PrescriptionNumber موږ ته Date, Drug, and Dosage مشخصوي یا یی Determent کوي که چیرته نسخه یواځي د یو کس لپاره وي نو بیا په دی صورت کی CustomerName, CustomerPhone, اوCustomerEmailAddress هم مشخصوي او PrescriptionNumber زموږ Candidate key بلل کیږي خو دا امکان نلري ځکه چي یو ډاکتر ته زیات ناروغان راځي

که چیرته دغه ټول فیلډونه په یو تیبل کي موجود وی نو یواځنی PrescriptionNumber دي چي ټول فیلډنه Determent کولاي شي

خو ستونزه دلته نه حل کیږي

Step 2 of the Normalization Process

- په دغه مرحله کی موږ functional dependencies مشخصوو
- په دي تیبل کی CustomerEmailAddress یو functional dependencies

کیداي شي

CustomerEmailAddress → (CustomerName, CustomerPhone)

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

Step 3 of the Normalization Process

- په دي مرحله کی موږ هغه Determent معلوموو چيCandidate key نشي کیدلای لکه CustomerEmailAddress
- په دي مرحله کی چي ستونزه له منځه یوسو تیبل په دوو برخو ویشوو

(c) ketabton.com: The Digital Library

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 ټيبل Normalize شو او ستونزه له منځه لاړه

نور مثالونه يی په سلايډونو کي کتلاي شی

The E-R Model (The Entity Relationship Model)

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

- **Entities**
- **Attributes**
- **Identifiers**
- **Relationships.**

E-R Modal موږ ته په ډيتابيس کی د Entity مجموعه رابنيي او د سيستم عناصر تشريح کوي او هغه دياګرام چي موږ يکي E-R Modal ډيزاين کوو هغه ته Entity-Relationship Diagram

يا E-R Diagram وايی

Entity: دا يواځي يو شخص نه ده او نه يو ځاي يا يو ابجيکټ ده هر هغه څه چی 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

- Person — EMPLOYEE, STUDENT, PATIENT
- Place — STORE, UNIVERSITY, STATE, COUNTRY
- Object — MACHINE, BUILDING, AUTOMOBILE
- Event — SALE, REGISTRATION, RENEWAL
- Concept — ACCOUNT, COURSE, WORKCENTER

Entity Class and Entity Instance

Entity Class: دEntity ګانو مجموعه چي يو ډول خواص ولري Entity Class بلل کيږي

(c) ketabton.com: The Digital Library

Entity Class موږ ته د Entity جوړښت هم په لاس راکوي چي دا کوم ډول Entity ده او هره Entity په E-R model کي یو نوم لري او Entity Class موږ په غټو حروفو لیکو مثال EMPLOYEE

Entity Instance: کله چي موږ د entity لپاره یو ټیبل جوړ کړو نو دي ټیبل کي هري entity ته که لږ وی او که زیات وی ټولو ته Entity Instance ویل کیږي

Strong Entity vs Weak Entity Types

Strong Entity Type: هغه entity ده چي په نورو entity پوري تړاو ونه لری او ازاد او خپلواک وی مثال: Student, employee, automobile, course

Weak Entity Type: هغه entity ده چی د هغي موجودیت په نورو entity پسي تړاو ولري او ازاد او خپلواک نه وی Grade, salary, model, subject

او هغه entity چي په هغه پوري نوري week entity تړاو ولری نو هغه ته **identifier owner** یا **owner** ویل کیږي

Attributes

هره entity یو یا څو attribute لري چي entity تشریح کوي

EmployeeName, DateOfHire, and JobSkillCode. مثال

Properties of Attributes: هر attribute ډیتاتایپ لري

- Character
- Numeric
- Date
- Currency, etc.

Types of Attributes

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

- Examples: CAR (ID, Color, Brand)

Composite Attribute: دا هغه attribute دي چي په برخو ویشل کیږي

(c) ketabton.com: The Digital Library

Example: Name, Address

Single Valued Attribute: دا هغه attribute دي چي single ویلو په ټیبل کي ذخیره کوي

Derived Attribute: دا هغه attribute دي چي value یې محاسبه کیږي کله age, total price

Identifier Attribute: دا هغه attribute دي چي په ټول ټیبل کی unique وی

Identifiers

هغه attribute چي د هغي پواسطه یوه entity مشخص کیدلی او پیژندل کیدلاي شي

Single Identifier: هغه attribute چي په یواځي ځان له تولي row څخه نمایندگی کولاي شي single identifier بلل کیږي کله ID

Composite Identifier: دا هغه identifier ده چي د دوه یا څو attributes د یوځاي کولو څّخه منځته راځی مثال: (FirstName, LastName, PhoneExtension).

Identifier کیداي شي unique وي یا non-unique وي

Unique Identifier: هغه identifier دي چي یواځي له یو entity instance څخه نمایندگي کوي مثال: Employee Number

Non-Unique Identifier: هغه identifier دي چي له یو سیټ څخه نمایندگي کوي

مثال: EmployeeName

Identifiers vs Keys

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

1. لومړي identifier فکري او خیالي مفهوم دي کیداي شي یو یا زیات attribute ولري
2. دوهم key به ضروري unique وي او Identifier کیداي شي unique وي یا-non unique وي

Relationships

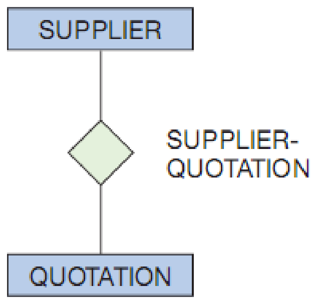
د دوو یا څو ټیبلونو لاجیکل تړاو ته Relationships وایي تر څو یو attributes له بل attributes سره تړاو ورکوي

Binary and Ternary Relationships: هغه Relationships چي له د دوو ټیبلونو تر منځ جوړشوي وی هغه ته دویمه درجه یا .binary relationships هم ویل کیږي

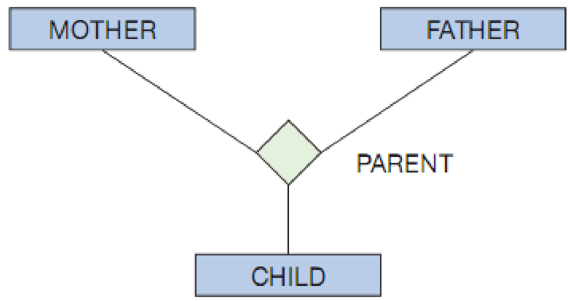
او که د دریو ټیبلونو تر منځ جوړ شوی وی هغه ته دریمه درجه یا .ternary relationships

هم ورته ویل کیږي

(c) ketabton.com: The Digital Library



(a) Binary Relationship



(b) Ternary Relationship

Types of Binary Relationships

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

په اسانه ډول سره داسي وايو (چې يو تيبل له بل تيبل سره يواځې په يو key سره ارتباط ولري one to one Relationships بلل کيږي)

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

يا هغه Relationships ته ويل کيږی چی د يو تيبل primary key د بل تيبل د foreign key سره connect وي

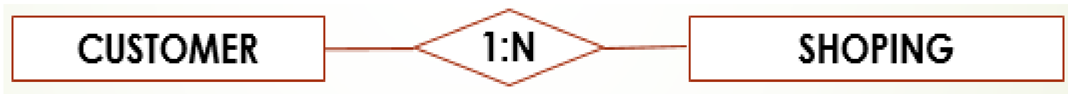
The many-to-many (N:M) relationship: هغه Relationships دي چي د زياتو تيبلونو instance له نورو زياتو تيبلونو instance سره ارتباط لري

Maximum and Minimum Cardinality

د Relationships درى ډوله چي مور يې مخکي يادونه وکړه دا د Cardinality پواسطه طبقه بندی شوي ده چي هدف يې شمير دي

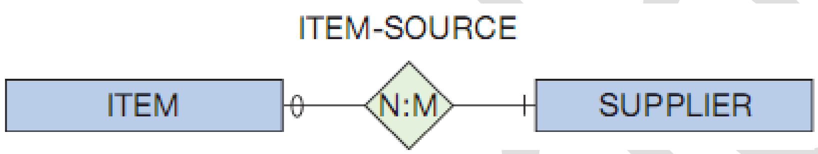
(c) ketabton.com: The Digital Library

Maximum Cardinality: هغه Cardinality ته ویل کیږي چي د یو entity زیات instance په یو Relationships برخه اخستلاي شی



Minimum Cardinality: هغه Cardinality ته ویل کیږي چي د یو entity کم instance په یو Relationships برخه اخستلاي شی

او که چیرته Cardinality صفر وی نو ضروري نه ده چي خامخا دی په Relationships برخه واخلي



SELECT & PROJECT Operations in Relational Algebra

The SELECT Operation: دا Operation ددي لپاره استعمالیږي چي څو فرعي ریکارډونه له یو تیبل څخه انتخاب کړو

SELECT operation یو filter دي چي یواځي هغه ریکارډونه را اخلي کوم چي له condition سره برابر وي هغه condition چي موږ ورته ورکړي وي او هغه ریکارډونه چي موږ ته بنیی په افقي ډول وی

select operation دوه برخي لري

1. هغه ریکارډونه چي له condition سره برابر وی هغه موږ بنیی
2. هغه ریکارډوه چي له condition سره برابر نه وی له هغو ریکارډونو څخه صرف نظر کیږي

دselect operation syntax په لاندي ډول ده

$\sigma_{\text{<selection condition>}}(R)$ په دی کی سګما د select معنی ورکوي

لومړي سګما علامه بیا condition لیکو او ورسته د هغه تیبل نوم لیکو له کوم تیبل نه چی ریکارډونه را اخلو.

کیدای شي چي شرطونه له یو زیات وي مثال په ډول

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

مثالونه: د EMPLOYEE په نامه تیبل څخه هغه ریکارډونه چي په 4 نمبر department کي کار کوي سلیکټ کړي؟

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

(c) ketabton.com: The Digital Library

هغه کارکوونکی راته انتخاب کړی چی معاش يی له 30000 څخه زيات وي

$\sigma_{Salary>30000}$ (EMPLOYEE)

The PROJECT Operation: دا operation له ټيبل څخه موږ ته کالمونه يا attribute انتخابوي او هغه کالمونه چي له condition سره برابر نه وي له هغه څخه صرف نظر کوی دا هم دوه برخی 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 وی برابر وي

Select operation Syntax

$\pi_{<projection\ condition>}$(R) په دغه کی د پای د **PROJECT** معنی لري

د هغو کارکوونکو نومونه، تخلص او معاش راته انتخاب کړی

$\pi_{name,\ Lname,\ salary}$(EMPLOYEE)

مثال: د کارکوونکو جنس او معاش راته انتخاب کړه $\pi_{gender,\ salary}$(EMPLOYEE)

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

The Set Operations in Relational Algebra

موږ به دلته دري عمليي تشريح کوو چي عبارت دي له union, intersection, او set difference (د سيټونو اتحاد، تقاطع او د سيټونو منفي عمليه) خو دلته د سيټونو څخه ټيبلونو مراد دي

او په دي U علامی سره ښوودل کيږي مثال : لومړي ټيبل U دوهم ټيبل

UNION: په دغه عمليه کي ټول هغه ريکارډونه چي په **R** او **S** ټيبلونو کي موجود وي يعني هغه ريکاډونه چي په **R** ټيبل کي موجود وي او په **S** په ټيبل کی هم موجود وی له دواړو څخه يو ريکارډ را اخيستل کيږي او په دريم ټيبل کي ځاي پرځاي کيږی او داسی ډول سره ښودل کيږي **R U S**

اتحاد په دي **U** علامی سره ښوودل کيږي مثال : لومړي ټيبل **U** دوهم ټيبل

(c) ketabton.com: The Digital Library

د STUDENT او INSTRCTOR دوه ټيبلونو اتحاد معلوم کړي

STUDENT ∪ 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 عمليه هغه ريکارډونه را اخيستل کيږی چی په دواړو ټيبلونو کی يوشان وي مثلاً يو ريکارډ په R ټيبل کي موجود ده او په S ټيبل کي هم موجود ده له دواړو ټيبلونو څخه همدغه يو ريکارډ را اخسيتل کيږي او له نورو څخه صرف نظر کيږی **R ∩ S,** تقاطع په دی ∩ علامي سره ښودل کيږي

د STUDENT او INSTRCTOR دوه ټيبلونو تقاطع معلوم کړي

STUDENT ∩ 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 عمليه کی هغه ريکارډونه ښودل کيږي چي په دواړو کی توپير ولري مثال په R او S ټيبلونو کي دوه ريکارډونه يوشان دي او نور ټول مختليف دي R – S عمليه کي دغه دوه ريکارډونه ختميږی نور ټول په دريم ټيبل کی ښودل کيږی او د سيټونو تفريق په (–) علامی باندي ښودل کيږي مثال: لومړي ــ دوهم ټيبل

R ∪ S = S ∪ R او **R ∩ S = S ∩ R** که لومړی هر ټيبل وليکو هيڅ فرق نه کوي

مګر Set difference کي **R – S ≠ S – R** نه دی سره مساوي

(c) ketabton.com: The Digital Library

د STUDENT او INSTRCTOR دوه ټیبلونو تفریق 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

(e)

Fname	Lname
John	Smith
Ricardo	Browne
Francis	Johnson

The Cartesian Product: دي ته CROSS PRODUCT یا CROSS JOIN هم ویل کیږي چي د دوو ټیبلونو د ریکارډونو ضرب عملیه ده او (X) سره ښودل کیږي

دا هم binary set عملیه ده مګر هغه ټیبلونه چي دغه عملیه وربانديي سرته رسوو باید د ټیبلونو عناصر مختلیف وي او ددي عملیی محصول د ټولو ټیبلونو ریکارډونه په نوي ټیبل کي ځاي پر ځاي کیږي

که چیرته د دوو ټیبلونو attributes یو ډول نومونه لري مثال ID او یا name په دواړو ټیبلونو کي موجود دی نو دلته ضرب عملیه نه ترسره کیږي سیستم دا نه پیژني چی دا اوس د student name ده که د customer name ده

دا ستونزه د ختمولو لپاره لومړی د ټیبل نوم او بیا ډاټ لیکو او وروسته attribute لیکو مثال:

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

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

(c) ketabton.com: The Digital Library

The JOIN Operation: په دغه عمليه کي له دوو تيبلونو څخه ډيټا يو تيبل ته رازي مثال يو تيبل کي دوه ريکارډونه دی او بل تيبل کي دري ريکارډونه دي دواړه چي سره شي پينځه ريکارډونه کيږي ادا عمليه په دغه ⋈ سره ښودل کيږي

که چيرته موږ وغواړو چي له دوو تيبلونو څخه معلومات را واخلو نو د **Join** عمليی سره په اسانی سره کولاي شو تر څو کار ترسره کړو

Type of Join Operation

1. **THETA JOIN:** په دغه کی ټول هغه ريکارډونه له دواړو تيبلونو څخه را اخلي کوم چي موږ condition ورته ورکړي وي مثال :

$$\mathcal{R}_1 \bowtie_{A\theta B} \mathcal{R}_2$$

تيتا **θ** به له Relation operator **{<,≤,=,>,≥}** څخه يو وي

2. **EQUIJOIN:** په دغه عمليه کي هر وخت د مساوي علامه کارول شوي وي

$$\mathcal{R}_1 \bowtie_{A=B} \mathcal{R}_2 = \sigma_{A=B}(\mathcal{R}_1 \times \mathcal{R}_2)$$

3. **NATURAL JOIN:** دغه Join هغه وخت ترسره کيږي چي د attribute نومونه يوشان وي په دغه حالت کي به د Rename عمليه ترسره کوي

مثال: **R 1 , R 2** تيبلونه څخه ادا **join X 1 ,X 2 ,X 3** کوو

(R1.X1 = R2.X1) ∧ (R1.X2 = R2.X2) ∧ (R1.X3 = R2.X3)

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*.	$\sigma_{\text{<selection condition>}}(R)$
PROJECT	Produces a new relation with only some of the attributes of *R*, and removes duplicate tuples.	$\pi_{\text{<attribute list>}}(R)$
THETA JOIN	Produces all combinations of tuples from R_1 and R_2 that satisfy the join condition.	$R_1 \bowtie_{\text{<join condition>}} R_2$
EQUIJOIN	Produces all the combinations of tuples from R_1 and R_2 that satisfy a join condition with only equality comparisons.	$R_1 \bowtie_{\text{<join condition>}} R_2$, OR $R_1 \bowtie_{(\text{<join attributes 1>}),(\text{<join attributes 2>})} R_2$
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.	$R_1 *_{\text{<join condition>}} R_2$, OR $R_1 *_{(\text{<join attributes 1>}),(\text{<join attributes 2>})} R_2$ OR $R_1 * R_2$

(c) ketabton.com: The Digital Library

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 $Z = X \cup Y$.	$R_1(Z) \div R_2(Y)$

منځنی ازمویني پوري د Fundamental of Database لند ليکچرنوټ پاي

Wiqade

(c) ketabton.com: The Digital Library

Wiqade

Get more e-books from www.ketabton.com
Ketabton.com: The Digital Library