RDBMS

An overview of Relational Databases

Symbiosis School for Liberal Arts

Shubhangi Hora

14060321085

DATABASE	An organized collection of data
RELATIONAL DATABASE	A database structured to recognize relations between stored items of information

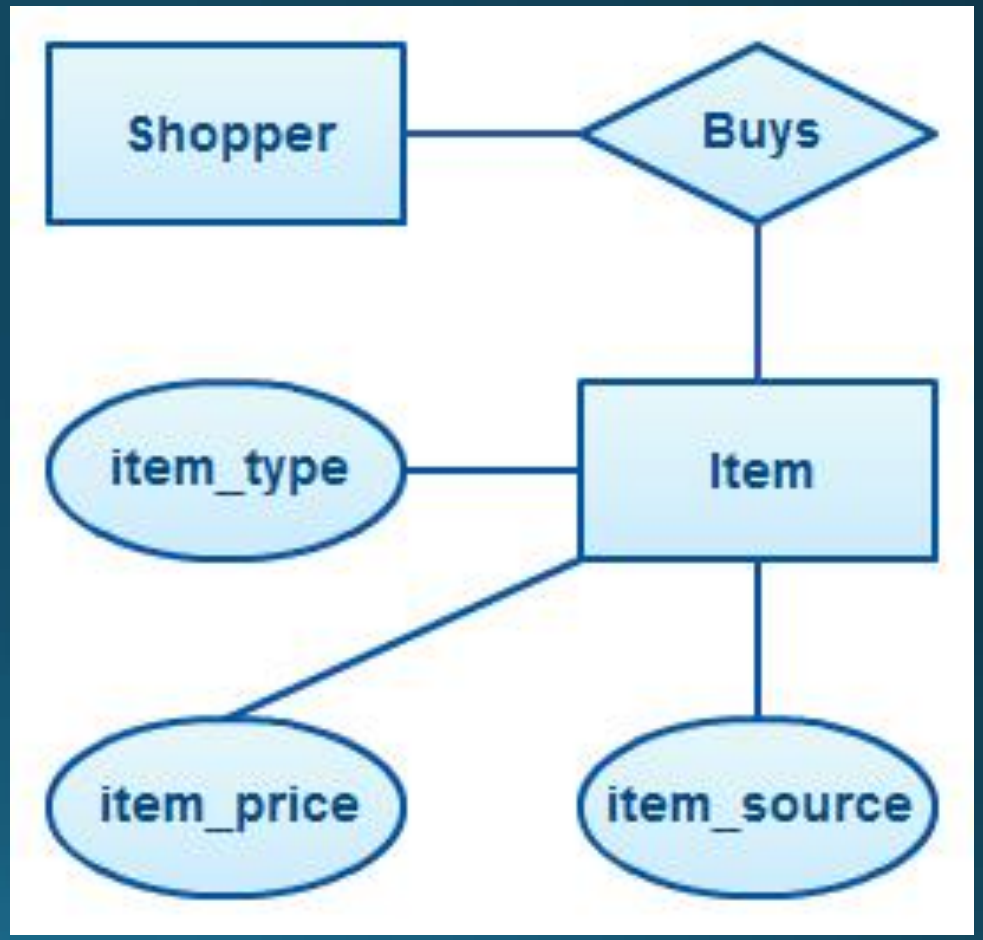
FILE SYSTEMS
Can't store large amounts of data
Can't access large amounts of data simultaneously
Complete programs are required to access and perform tasks on data
No concurrency
Data can't be restored to a consistent state if the system crashes
Very little security

DATABASES

1. Data independency
2. Efficient data access
3. Data integrity and security
4. Data administration
5. Concurrent access and crash recovery

1. Requirements Analysis
2. Conceptual database design
3. Logical database design
4. Schema refinement
5. Physical database design
6. Security design

Entity – Relationship Diagram



- Entity – an object that is distinguishable from other objects
- Entity set – a set of objects that possess some similarities but are distinguishable from other objects
- Attributes – aspects used to describe an entity
- Domain – possible values of an attribute
- Primary key – a value of an attribute that uniquely identifies an entity from its entity set
- Relationship – an association between two or more entities
- Relationship set – a set of similar relationships

Relational operator – a construct that defines a relation between two entities

Union compatible

- Union: displays all tuples that occur in either R or S
- Intersection: displays tuples that occur in both R and S
- Difference: R – S returns all tuples that occur in R but not S
- Cartesian Product: R x S returns ordered pairs of tuples, wherein the first tuple is from R and the second is from S

- Selection: a horizontal subset of tuples for a particular condition
- Projection: a vertical subset of columns for particular attributes
- Join: cross product of tables followed by selections/ projections
  - Inner joins – includes tuples which have matching attributes
  - Outer joins – include all tuples
- Division: reverse of Cartesian product

R

Name	Cust_status
Shubhangi	Good
Pooja	Bad

S

Name	Cust_status
Shubhangi	Good
Nita	Excellent

Select * from R where Cust_status = Good

Name	Cust_status
Shubhangi	Good

Select Name from R

Name
Shubhangi
Pooja

Select Name from R, S where R.Cust_status = S.Cust_status

Name
Shubhangi

Normalization – a technique to organize tables to avoid redundancy, dependency and anomalies of data

Student Table			
S_id	**S_name**	**Salutation**	**Subjects**
12	Shubhangi	Ms.	Biology, Drama
13	Pooja	Ms.	Economics

1. Functional dependency – an attribute uniquely identifies another attribute
2. Partial dependency – an attribute that is part of a multipart candidate key uniquely identifies another attribute
3. Transitive dependency – changing a non-key attribute may require a change in another non-key attribute

Student Table			
S_id	S_name	Salutation	Subjects
12	Shubhangi	Ms.	Biology
12	Shubhangi	Ms.	Drama
13	Pooja	Ms.	Economics

Student Table		
S_id	S_name	Salutation
12	Shubhangi	Ms.
13	Pooja	Ms.

Subject Table	
S_id	Subjects
12	Biology
12	Drama
13	Economics

Student Table	
S_id	S_name
12	Shubhangi
13	Pooja

Subject Table	
S_id	Subjects
12	Biology
12	Drama
13	Economics

Salutation Table	
S_id	Salutation
12	Ms.
13	Ms.

SQL – Structured Query-based Language

1. Data Declaration Language (DDL) – commands used to define the structure of your schema

Eg: CREATE table

2. Data Manipulation Language (DML) – commands used to manipulate your data

Eg: INSERT, SELECT table

Customer	
C_id	**Name**
1	Shubhangi
2	Pooja

Order	
O_id	**C_id**
104	1

Select Customer.C_id, Customer.Name, Order.O_id from Customer Left Join Order On Customer.C_id = Order.C_id

C_id	Name	O_id
1	Shubhangi	104
2	Pooja	*null*

Select Customer.C_id, Customer.Name, Order.O_id from Order Right Join Customer On Customer.C_id = Order.C_id

C_id	Name	O_id
1	Shubhangi	104
2	Pooja	*null*

Concurrency Control – techniques to coordinate simultaneous transactions while preserving data integrity

Pessimistic – an entity is locked the entire time it exists in the database's memory

Read (shared) and write (exclusive) locks

Strict Two – Phase Locking:

1. If a transaction wants to read an object, it acquires a shared lock and if it wants to modify an object, it acquires an exclusive lock
2. All locks held by a transaction are released when the transaction is completed

T1	T2	T3	Locks
S(A)			Granted
R(A)			
	X(A)		Denied – T2 blocked
	W(A)		
	X(B)		Granted when T2 is unblocked
	W(B)		
		X(B)	Granted
		W(B)	
X(B)			Denied – T1 blocked
W(B)			
Com.	Com.	Com.	All acquired locks released

Optimistic – doesn't prevent, rather waits for the conflicts to occur and resolves them

Timestamps are applied to transactions before they enter the validation phase

1. Ti completes (all three phases) before j begins; or
2. Ti completes before Tj starts its Write phase, and Ti does not write any database object that is read by Tj;
3. Ti completes its Read phase before Tj completes its Read phase, and Ti does not write any database object that is either read or written by Tj.

Query Optimization – determining the most efficient way to execute a query

1. Pipelined Evaluation – a query composed of several operators, **pipelines** the result of one operator to another
2. Pushing selections – performing selections before the join occurs
3. Using indexes

CAP theorem

Consistency

Availability

Partition tolerance

Cannot have all three

Crash Recovery – preparing for and recovering from hardware, software, network, process or system failure

Ensures atomicity and durability

Undoes actions of transactions that haven't committed

Ensures actions of committed transactions survive system crashes and media failures

- User failure
- Statement failure
- Process failure
- Network failure