

Database Administration

Lecture 05: Performance Monitoring and Tuning.

Ciolli et al. and Grafana Labs.

24 de febrero de 2025

Database Administration: Reverse Design.

Udemy Explorar Buscar cualquier cosa

Desarrollo > Diseño y desarrollo de bases de datos > Sistemas de gestión de bases de datos (DBMS)

The screenshot shows a course page for 'Database Design and Management' on Udemy. At the top, there's a navigation bar with 'Udemy', 'Explorar', a search bar, and account links for 'Udemy Business', 'Instructor', 'Mi aprendizaje', and social media icons. Below the navigation is a breadcrumb trail: 'Desarrollo > Diseño y desarrollo de bases de datos > Sistemas de gestión de bases de datos (DBMS)'. The main title 'Database Design and Management' is displayed with a play button icon. A large image shows a central blue cylinder representing a database with lines connecting it to several smaller orange rectangles representing tables or data structures. Below the image is a purple button labeled 'Vista previa de este curso'.

Database Design and Management

Learn how to design and manage database with ERD, database generation and reversal with Visual Paradigm.

Tutorial gratuito 4.5 ★★★★ 0 (3458 calificaciones) 95.409 estudiantes

12 h 41 min de video bajo demanda

Creado por [Visual Paradigm](#)

Idiomas: Inglés, Inglés [automático]

Ir al curso

Lo que aprenderás Contenido del curso Valoraciones Instructores

Requisitos

- Basic software development concept
- PC or OSX or Linux

Contenido del curso

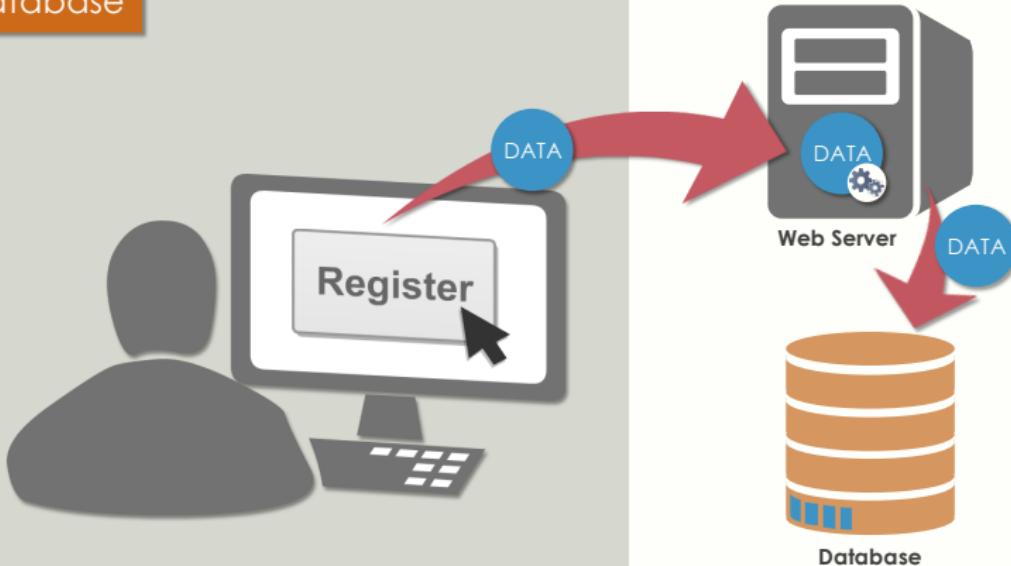
4 secciones • 14 clases • 2 h 41 m de duración total Ampliar todas las secciones

▼ Introduction to Database Design and Management	8 clases • 1 h 31 min
▼ Database Engineering	3 clases • 37 min
▼ Java Database Programming	1 clases • 17 min
▼ Best Practices	2 clases • 17 min

Content has been extracted from *Database Design and Management*. Udemy Course, created by Visual Paradigm, 2025. Visit <https://www.udemy.com/course/database-design-and-management/> and <https://www.visual-paradigm.com/>.

Plan

Database



Introduction

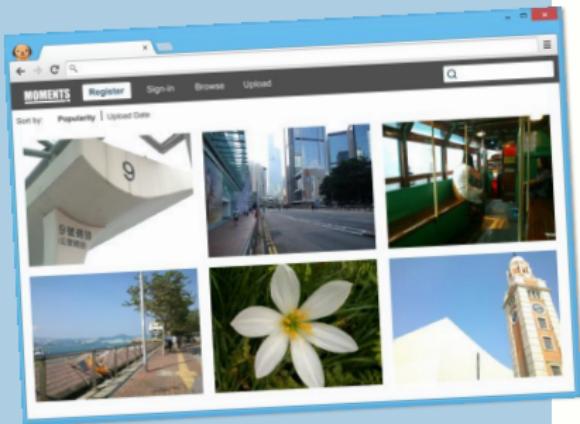
Database



Database stores
long-term persistent data

Introduction

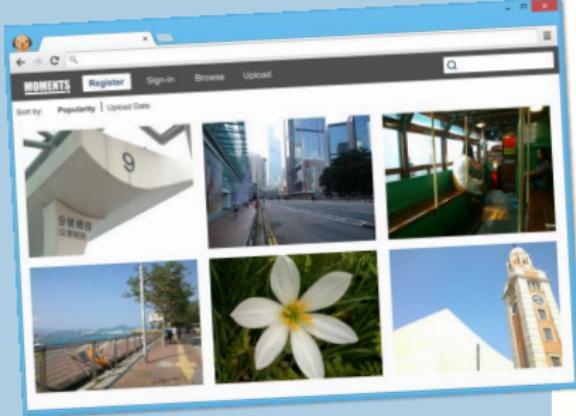
Database



Example:
MOMENTS online
photo album

Introduction

Database



Persistent data:

- Photos
- Uploaders info.
- Albums
- Upload history
- Comments

Introduction

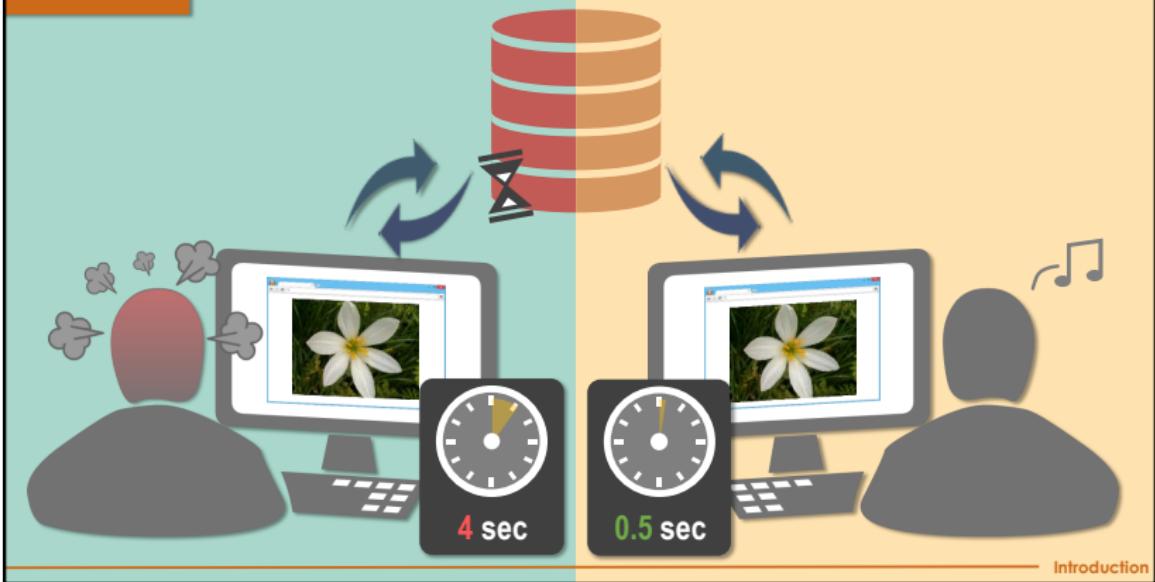
Database



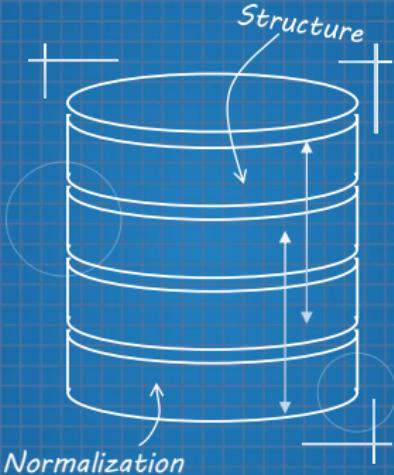
Database performance
influences
software performance

Introduction

Database



Database Design



Database is a blueprint of
your **persistent structure**

Introduction

Database Design - Why



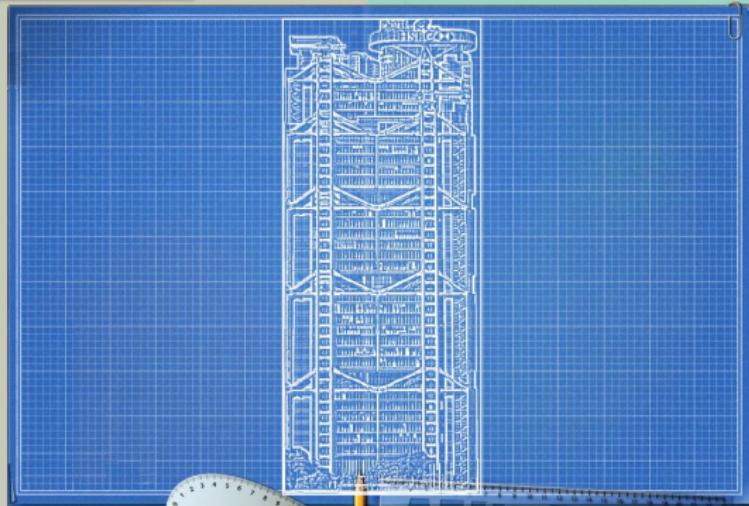
Introduction

Database Design - Why



Introduction

Database Design - Why



Introduction

Database Design - Why

```
My Tic-Tac-Toe results:  
Game 1: Win  
Game 2: Win  
Game 3: Draw  
Game 4: Lost  
Game 5: Win  
  
Press any key to continue...
```



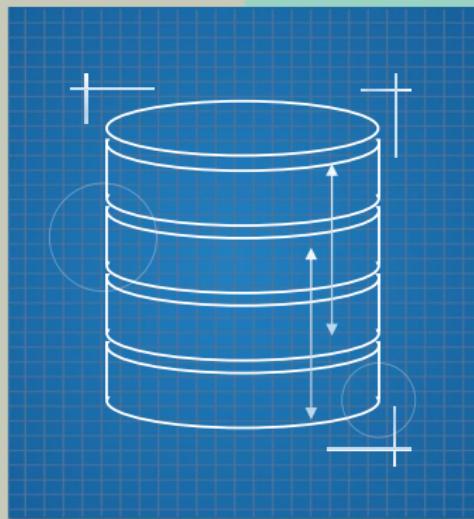
Introduction

Database Design - Why



Introduction

Database Design - Why



Introduction

Database Design – Goal



Maintain a balance
between **DB size** &
performance

Introduction

Database Design – Goal



How **uploader's** details
are stored?

Introduction

Database Design – Goal

Photo and Uploader details in one table

ID	Title	Uploader Name	Uploader PhoneNum	Uploader Address
1	The tram	Chris Wong	252-608-5628	572 Lost Lake Crossing
2	Little kitten	Chris Wong	252-608-5628	572 Lost Lake Crossing
3	Chamomile	Chris Wong	252-608-5628	572 Lost Lake Crossing
4	New bus route	Wanda A. Allison	808-839-2217	2017 Green Impasse
5	Ferry pier 9	Chris Wong	252-608-5628	572 Lost Lake Crossing
6	The coast	Chris Wong	252-608-5628	572 Lost Lake Crossing
7	Rock climbing	Wanda A. Allison	808-839-2217	2017 Green Impasse
8	Fountain	Chris Wong	252-608-5628	572 Lost Lake Crossing
9	Sunset	Wanda A. Allison	808-839-2217	2017 Green Impasse

Database table: **Photo**



DB Size
Performance

Introduction

Database Design – Goal

Access both **photo** and **uploader** details
when querying a single record

ID	Title	Uploader Name	Uploader PhoneNum	Uploader Address
1	The tram	Chris Wong	252-608-5628	572 Lost Lake Crossing
2	Little kitten	Chris Wong	252-608-5628	572 Lost Lake Crossing
3	Chamomile	Chris Wong	252-608-5628	572 Lost Lake Crossing
4	New bus route	Wanda A. Allison	808-839-2217	2017 Green Impasse
5	Ferry pier 9	Chris Wong	252-608-5628	572 Lost Lake Crossing
6	The coast	Chris Wong	252-608-5628	572 Lost Lake Crossing
7	Rock climbing	Wanda A. Allison	808-839-2217	2017 Green Impasse
8	Fountain	Chris Wong	252-608-5628	572 Lost Lake Crossing
9	Sunset	Wanda A. Allison	808-839-2217	2017 Green Impasse

Database table: **Photo**



DB Size
Performance

Introduction

Database Design – Goal

However, this may create a lot of **redundant** data

ID	Title	Uploader Name	Uploader PhoneNum	Uploader Address
1	The tram	Chris Wong	252-608-5628	572 Lost Lake Crossing
2	Little kitten	Chris Wong	252-608-5628	572 Lost Lake Crossing
3	Chamomile	Chris Wong	252-608-5628	572 Lost Lake Crossing
4	New bus route	Wanda A. Allison	808-839-2217	2017 Green Impasse
5	Ferry pier 9	Chris Wong	252-608-5628	572 Lost Lake Crossing
6	The coast	Chris Wong	252-608-5628	572 Lost Lake Crossing
7	Rock climbing	Wanda A. Allison	808-839-2217	2017 Green Impasse
8	Fountain	Chris Wong	252-608-5628	572 Lost Lake Crossing
9	Sunset	Wanda A. Allison	808-839-2217	2017 Green Impasse

Database table: **Photo**



DB Size
Performance

Introduction

Database Design – Goal

ID	Title	Uploader ID
1	The tram	1
2	Little kitten	1
3	Chamomile	1
4	New busroute	2
5	Ferry pier 9	1
6	The coast	1
7	Rock climbing	2
8	Fountain	1
9	Sunset	2

Database table: **Photo**

ID	Name	PhoneNum	Address
1	Chris Wong	252-608-5628	572 Lost Lake Crossing
2	Wanda A. Allison	808-839-2217	2017 Green Impasse

Database table: **Uploader**



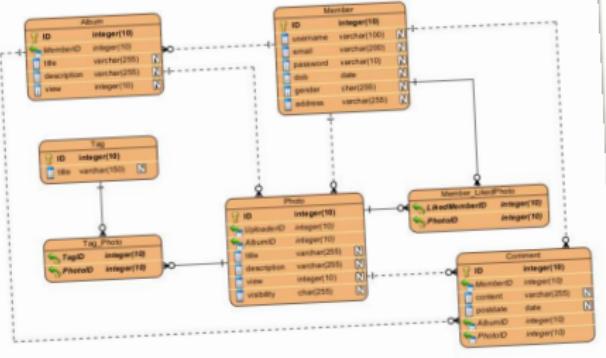
Photo and Uploader details
in **separate tables**



DB Size
Performance

Introduction

Database Design Notations



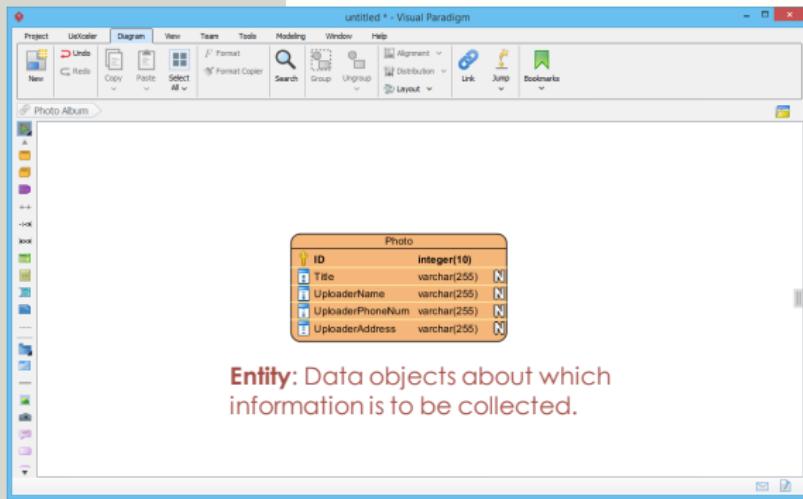
Entity Relationship Diagram (ERD)

Entity Relationship Diagram is:

- Language for expressing database
- Visual language
- Facilitate communication
- Presents data structure & inter-relationships

Introduction

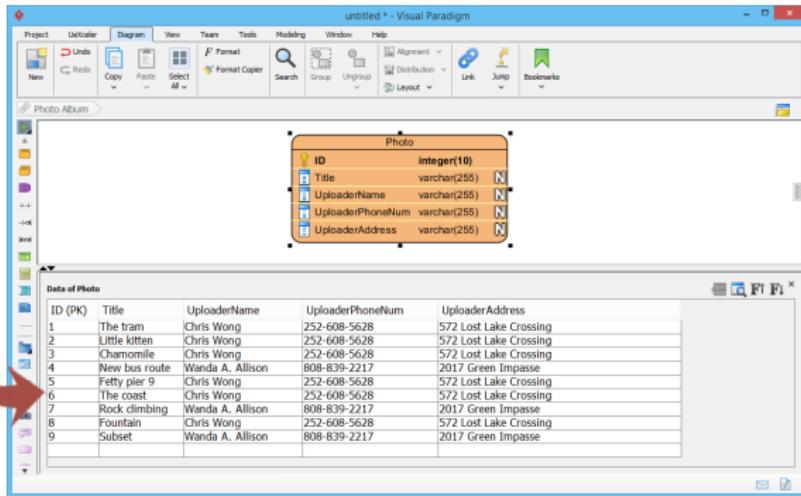
Database Design Notations



Entity: Data objects about which information is to be collected.

Introduction

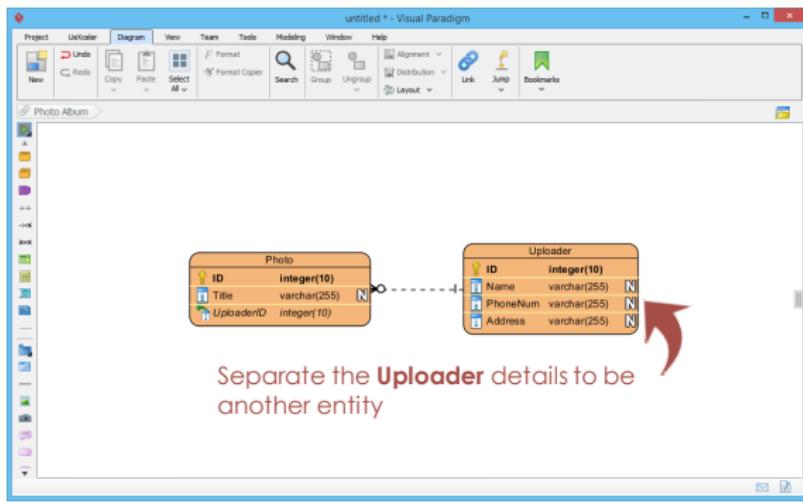
Database Design Notations



Records of
Photo table

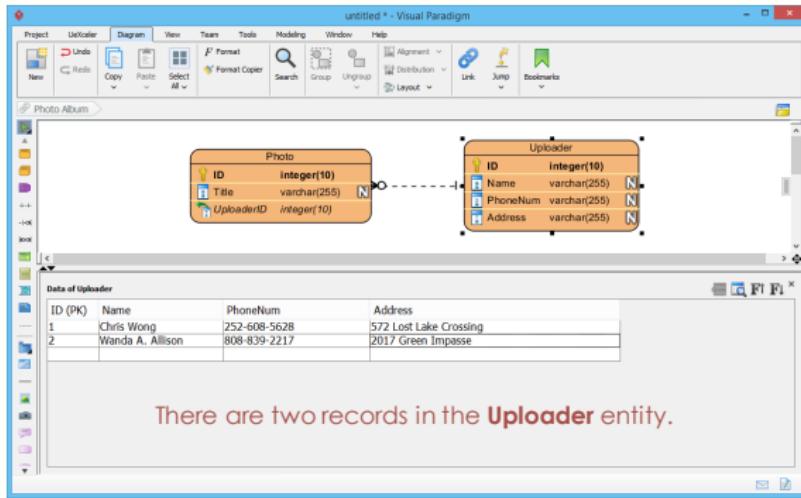
Introduction

Database Design Notations



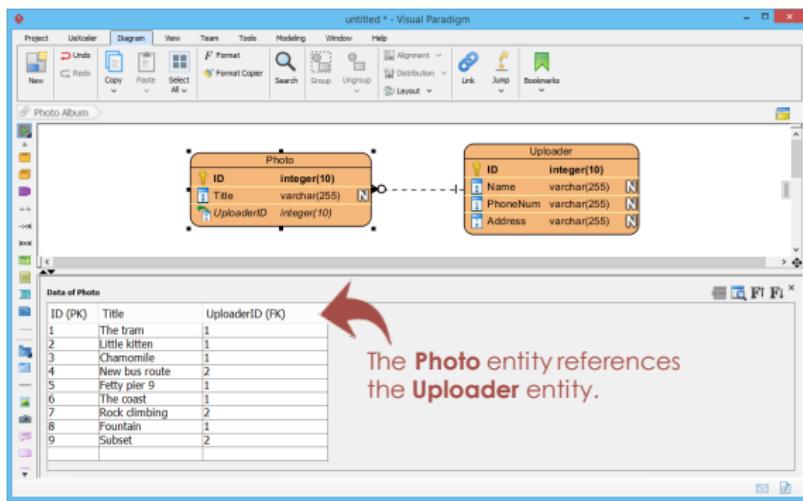
Introduction

Database Design Notations



Introduction

Database Design Notations



Introduction

Plan

What is entity and column?

- **Fundamental** elements of database design



Entity and Column

What is entity and column?

- **Fundamental** elements of database design
- Define how data are being **stored**



Entity and Column

What is entity and column?

- **Fundamental** elements of database design
- Define how data are being **stored**
 - E.g. Everything put together? Separate places for different data?



Entity and Column

What is entity and column?

- **Fundamental** elements of database design
- Define how data are being **stored**
 - E.g. Everything put together? Separate places for different data?
- Generalized form of data



Generalizing entity from data

Photo ID	Photo Title	Uploader Name	Uploader PhoneNum	Uploader Address
1	The tram	Chris Wong	252-608-5628	572 Lost Lake Crossing
2	Little kitten	Chris Wong	252-608-5628	572 Lost Lake Crossing
3	Chamomile	Chris Wong	252-608-5628	572 Lost Lake Crossing
4	New bus route	Wanda A. Allison	808-839-2217	2017 Green Impasse
5	Ferry pier 9	Chris Wong	252-608-5628	572 Lost Lake Crossing
6	The coast	Chris Wong	252-608-5628	572 Lost Lake Crossing
7	Rock climbing	Wanda A. Allison	808-839-2217	2017 Green Impasse
8	Fountain	Chris Wong	252-608-5628	572 Lost Lake Crossing
9	Sunset	Wanda A. Allison	808-839-2217	2017 Green Impasse

Photo Details

Entity and Column

Generalizing entity from data

Photo ID	Photo Title	Uploader Name	Uploader PhoneNum	Uploader Address
1	The tram	Chris Wong	252-608-5628	572 Lost Lake Crossing
2	Little kitten	Chris Wong	252-608-5628	572 Lost Lake Crossing
3	Chamomile	Chris Wong	252-608-5628	572 Lost Lake Crossing
4	New bus route	Wanda A. Allison	808-839-2217	2017 Green Impasse
5	Ferry pier 9	Chris Wong	252-608-5628	572 Lost Lake Crossing
6	The coast	Chris Wong	252-608-5628	572 Lost Lake Crossing
7	Rock climbing	Wanda A. Allison	808-839-2217	2017 Green Impasse
8	Fountain	Chris Wong	252-608-5628	572 Lost Lake Crossing
9	Sunset	Wanda A. Allison	808-839-2217	2017 Green Impasse

Photo

Entity and Column

Generalizing entity from data

The diagram illustrates the process of generalizing a database table into an entity. On the left, a table with columns 'Photo ID', 'Photo Title', 'Uploader Name', 'Uploader PhoneNum', and 'Uploader Address' is shown. Two red arrows point from the first two columns ('Photo ID' and 'Photo Title') to a box on the right labeled 'Photo'. This box contains two attributes: 'ID' and 'Title'. The other three columns ('Uploader Name', 'Uploader PhoneNum', and 'Uploader Address') are part of the 'Photo' entity but are not explicitly highlighted.

Photo ID	Photo Title	Uploader Name	Uploader PhoneNum	Uploader Address
1	The tram	Chris Wong	252-608-5628	572 Lost Lake Crossing
2	Little kitten	Chris Wong	252-608-5628	572 Lost Lake Crossing
3	Chamomile	Chris Wong	252-608-5628	572 Lost Lake Crossing
4	New bus route	Wanda A. Allison	808-839-2217	2017 Green Impasse
5	Ferry pier 9	Chris Wong	252-608-5628	572 Lost Lake Crossing
6	The coast	Chris Wong	252-608-5628	572 Lost Lake Crossing
7	Rock climbing	Wanda A. Allison	808-839-2217	2017 Green Impasse
8	Fountain	Chris Wong	252-608-5628	572 Lost Lake Crossing
9	Sunset	Wanda A. Allison	808-839-2217	2017 Green Impasse

Photo

ID
Title

Entity and Column
Entity and Column

Generalizing entity from data

Integer

ID	Title	Uploader Name	Uploader PhoneNum	Uploader Address
1	The tram	Chris Wong	252-608-5628	572 Lost Lake Crossing
2	Little kitten	Chris Wong	252-608-5628	572 Lost Lake Crossing
3	Chamomile	Chris Wong	252-608-5628	572 Lost Lake Crossing
4	New bus route	Wanda A. Allison	808-839-2217	2017 Green Impasse
5	Ferry pier 9	Chris Wong	252-608-5628	572 Lost Lake Crossing
6	The coast	Chris Wong	252-608-5628	572 Lost Lake Crossing
7	Rock climbing	Wanda A. Allison	808-839-2217	2017 Green Impasse
8	Fountain	Chris Wong	252-608-5628	572 Lost Lake Crossing
9	Sunset	Wanda A. Allison	808-839-2217	2017 Green Impasse

5 digits long

ID : integer (5)
Title

Entity and Column

Generalizing entity from data

Photo ID	Text	Uploader Name	Uploader PhoneNum	Uploader Address
1	The tram	Chris Wong	252-608-5628	572 Lost Lake Crossing
2	Little kitten	Chris Wong	252-608-5628	572 Lost Lake Crossing
3	Chamomile	Chris Wong	252-608-5628	572 Lost Lake Crossing
4	New bus route	Wanda A. Allison	808-839-2217	2017 Green Impasse
5	Ferry pier 9	Chris Wong	252-608-5628	572 Lost Lake Crossing
6	The coast	Chris Wong	252-608-5628	572 Lost Lake Crossing
7	Rock climbing	Wanda A. Allison	808-839-2217	2017 Green Impasse
8	Fountain	Chris Wong	252-608-5628	572 Lost Lake Crossing
9	Sunset	Wanda A. Allison	808-839-2217	2017 Green Impasse

Photo

ID : integer (5)
Title : varchar (120)

Entity and Column

Generalizing entity from data

Photo ID	Photo Title	Uploader Name	Uploader PhoneNum	Uploader Address
1	The tram	Chris Wong	252-608-5628	572 Lost Lake Crossing
2	Little kitten	Chris Wong	252-608-5628	572 Lost Lake Crossing
3	Chamomile	Chris Wong	252-608-5628	572 Lost Lake Crossing
4	New bus route	Wanda A. Allison	808-839-2217	2017 Green Impasse
5	Ferry pier 9	Chris Wong	252-608-5628	572 Lost Lake Crossing
6	The coast	Chris Wong	252-608-5628	572 Lost Lake Crossing
7	Rock climbing	Wanda A. Allison	808-839-2217	2017 Green Impasse
8	Fountain	Chris Wong	252-608-5628	572 Lost Lake Crossing
9	Sunset	Wanda A. Allison	808-839-2217	2017 Green Impasse

Photo

ID : integer (5)
Title : varchar (120)

Uploader

Entity and Column

Generalizing entity from data

Photo ID	Photo Title	Uploader Name	Uploader PhoneNum	Uploader Address
1	The tram	Chris Wong	252-608-5628	572 Lost Lake Crossing
2	Little kitten	Chris Wong	252-608-5628	572 Lost Lake Crossing
3	Chamomile	Chris Wong	252-608-5628	572 Lost Lake Crossing
4	New bus route	Wanda A. Allison	808-839-2217	2017 Green Impasse
5	Ferry pier 9	Chris Wong	252-608-5628	572 Lost Lake Crossing
6	The coast	Chris Wong	252-608-5628	572 Lost Lake Crossing
7	Rock climbing	Wanda A. Allison	808-839-2217	2017 Green Impasse
8	Fountain	Chris Wong	252-608-5628	572 Lost Lake Crossing
9	Sunset	Wanda A. Allison	808-839-2217	2017 Green Impasse

Photo

ID : integer (5)
Title : varchar (120)

Uploader

Name : varchar (255)
PhoneNum : varchar (20)
Address : varchar (255)

Entity and Column

Generalizing entity from data

Photo
ID : integer (5)
Title : varchar (120)

Uploader
Name : varchar (255)
PhoneNum : varchar (20)
Address : varchar (255)



Entity and Column

Generalizing entity from data

Photo

ID : integer (5)
Title : varchar (120)

Album

Title : varchar (120)
Description : varchar (255)
View : integer (10)

Comment

Content : varchar (255)
Postdate: date

Uploader

Name : varchar (255)
PhoneNum : varchar (20)
Address : varchar (255)

Location

Name : varchar (255)
Shortname : varchar (50)

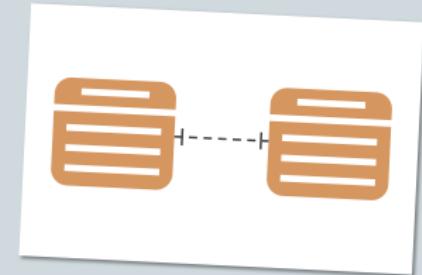
Tag

Title : varchar (120)
Description : varchar (255)

Entity and Column

What is ERD?

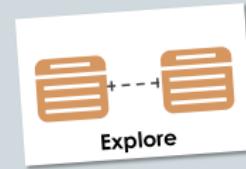
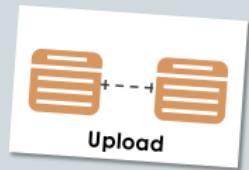
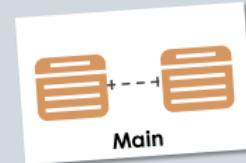
- A graph of entities and their relationships



Entity and Column

What is ERD?

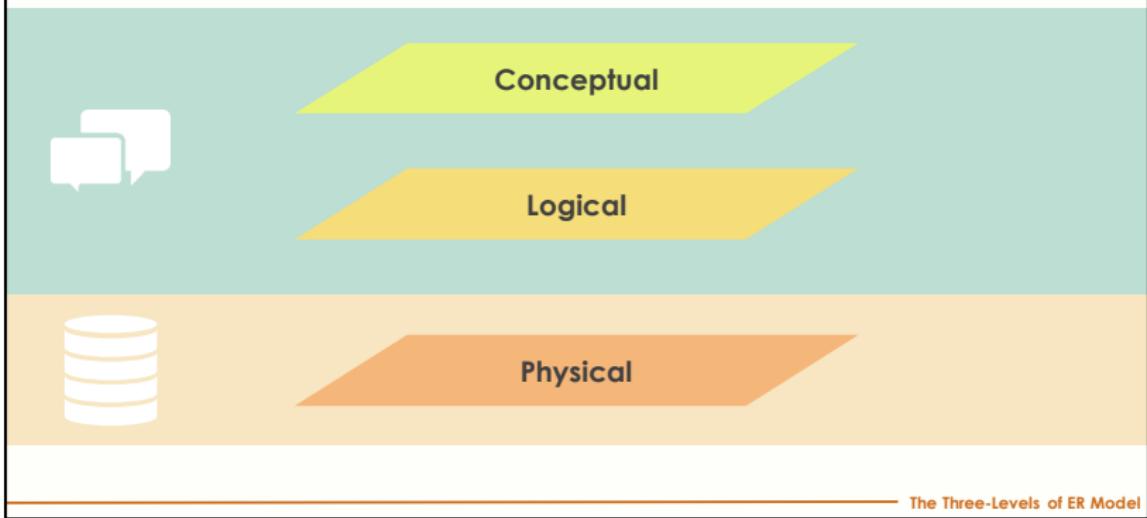
- A graph of entities and their relationships
- Multiple ERDs for different contexts



Entity and Column

Plan

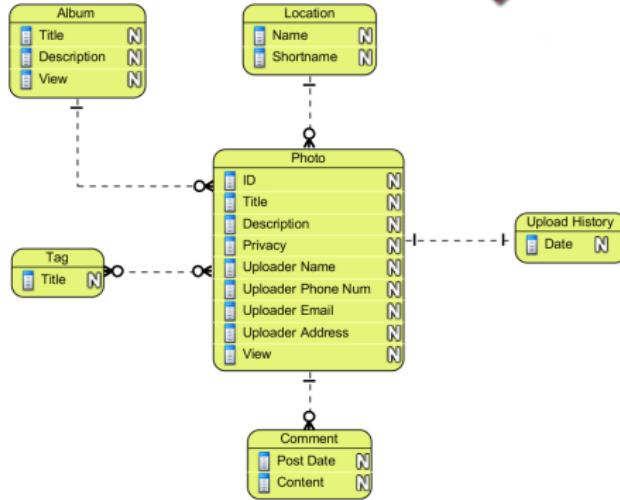
The levels



Example

Conceptual

Only main concepts and relationships are included

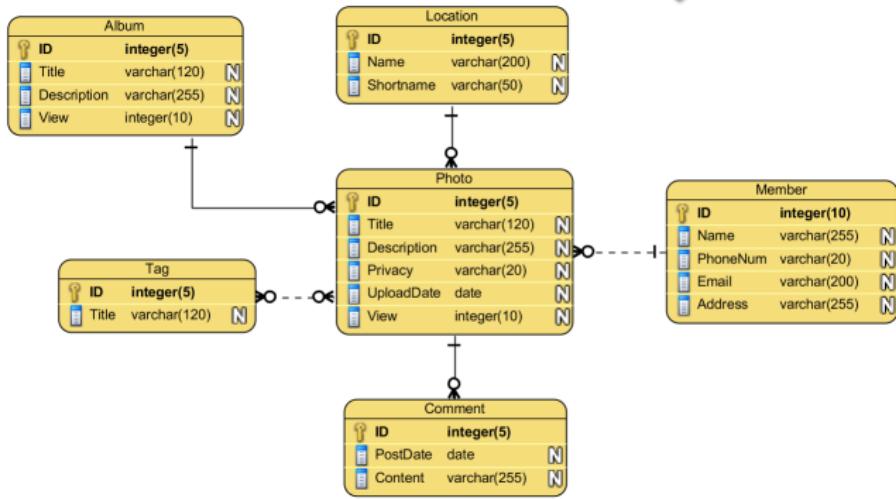


The Three-Levels of ER Model

Example

Logical

With detailed representation
of organization's data

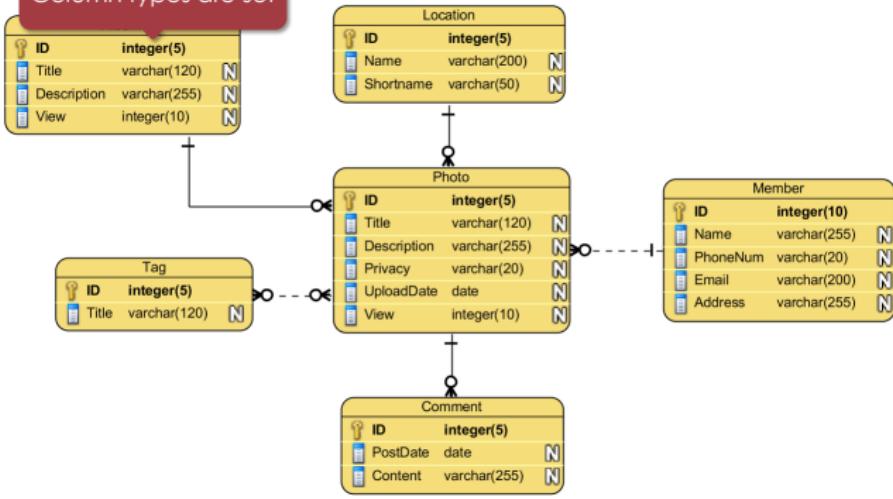


The Three-Levels of ER Model

Example

Logical

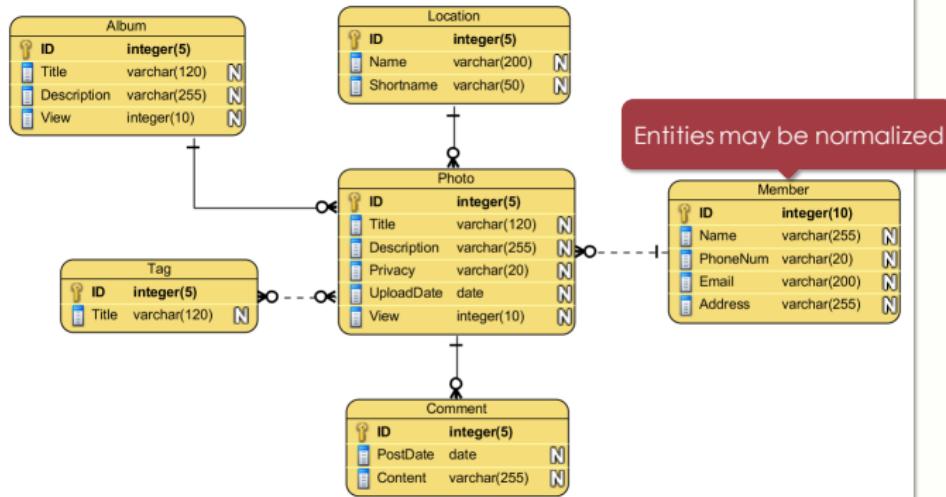
Column types are set



The Three-Levels of ER Model

Example

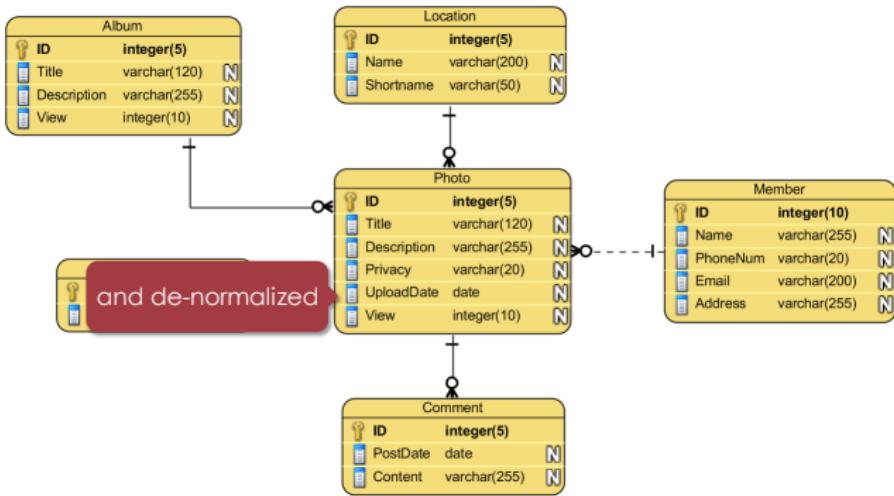
Logical



The Three-Levels of ER Model

Example

Logical

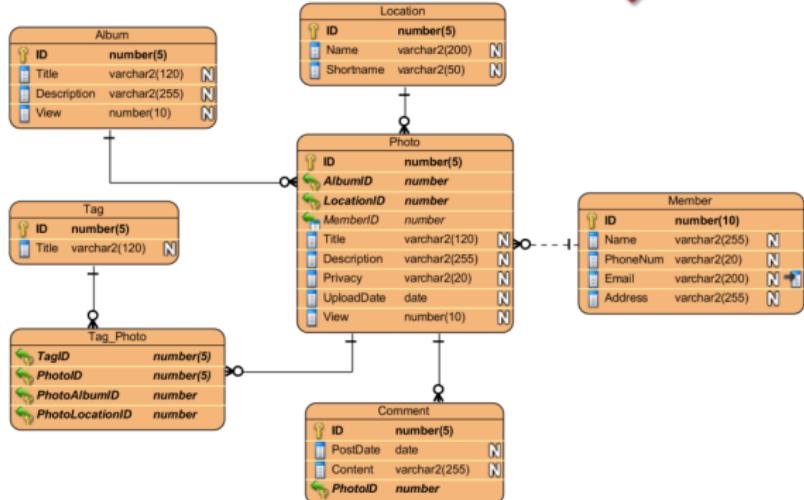


The Three-Levels of ER Model

Example

Physical

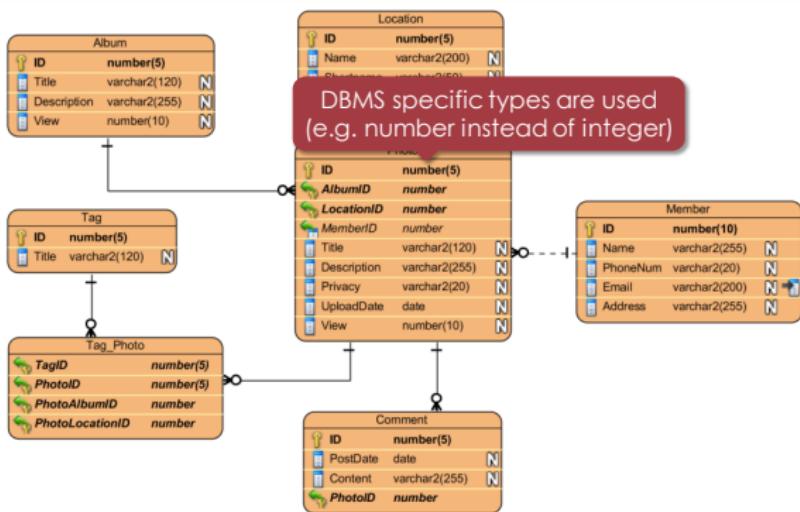
Designed for database creation/maintenance



The Three-Levels of ER Model

Example

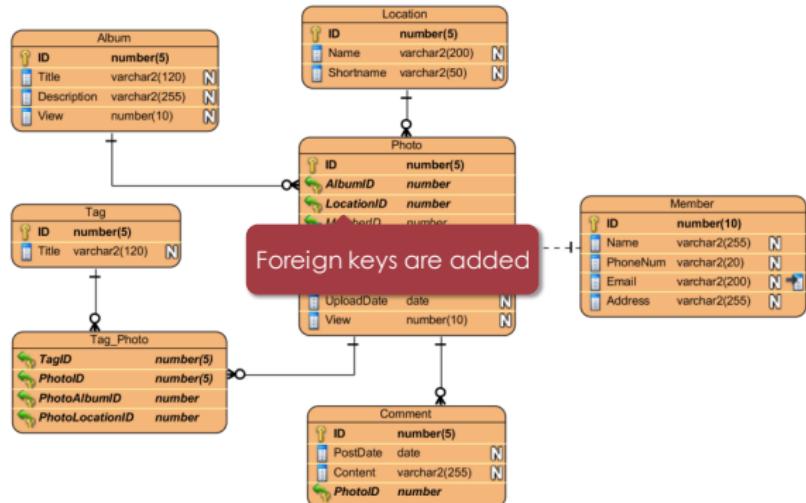
Physical



The Three-Levels of ER Model

Example

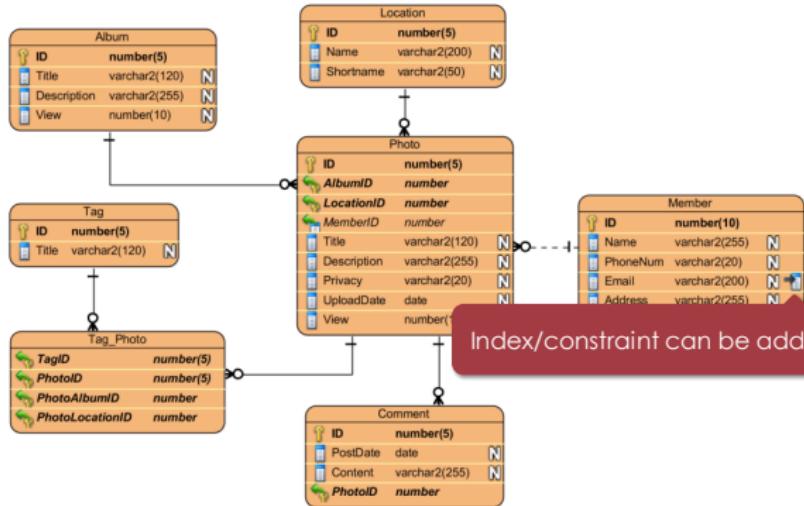
Physical



The Three-Levels of ER Model

Example

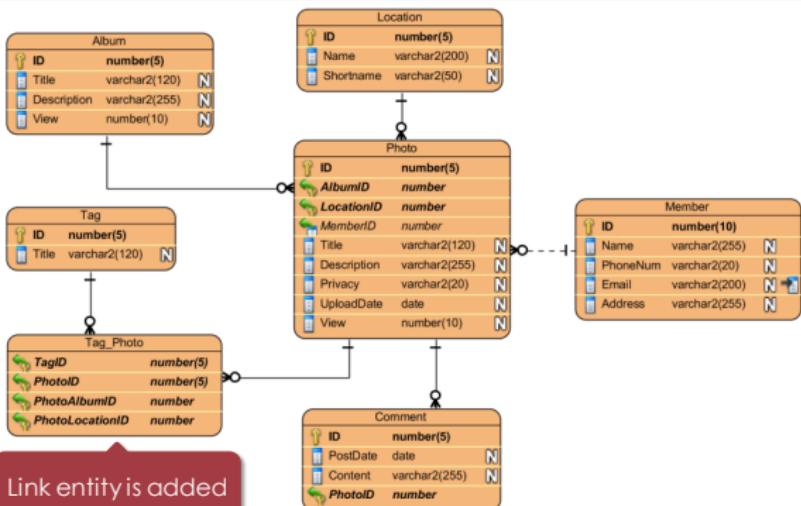
Physical



The Three-Levels of ER Model

Example

Physical



The Three-Levels of ER Model

Plan



Primary Key

Key and Relationship

Primary key is a table column
(or columns) which **uniquely** identifies
each record in a table.

Key and Relationship

What is Primary Key?

Member Name	Member PhoneNum	Member Address
Nancy Ward	620-824-6226	213 Cotton Valley
Nancy Ward	562-424-1087	843 Lost Village
Nancy Ward	719-427-0417	43 Noble Lane
Wanda A. Allison	713-728-7073	123 North Point
Peter R. Webster	216-317-2998	972 Golden Valley
Lucille A. Cola	662-519-0717	785 The Corner
Paul J. Clark	484-325-1544	125Good View

Table: Member

Key and Relationship

What is Primary Key?

Member Name	Member PhoneNum	Member Address
Nancy Ward	620-824-6226	213 Cotton Valley
Nancy Ward	562-124-1087	842 Lost Village
Nancy		Lane
Wanda		Point
Peter R. Webster	216-317-2998	972 Golden Valley
Lucille A. Cola	662-519-0717	785 The Corner
Paul J. Clark	484-325-1544	125Good View

What is the phone number
of **Nancy Ward**?

Table: Member

Key and Relationship

What is Primary Key?

Member Name	Member PhoneNum	Member Address
Nancy Ward	620-824-6226	213 Cotton Valley
Nancy Ward	562-424-1087	843 Lost Village
Nancy Ward	719-427-0417	43 Noble Lane
	713-728-7073	123 North Point
	216-317-2998	972 Golden Valley
Lucille A. Cola	662-519-0717	785 The Corner
Paul J. Clark	484-325-1544	125Good View

Which Nancy Ward do you mean?

Table: Member

Key and Relationship

What is Primary Key?

ID	Member Name	Member PhoneNum	Member Address
1	Nancy Ward	620-824-6226	213 Cotton Valley
2	Nancy Ward	562-424-1087	843 Lost Village
3	Nancy Ward	719-427-0417	43 Noble Lane
4	Wanda A. Allison	713-728-7073	123 North Point
5	Peter R. Webster	216-317-2998	972 Golden Valley
6	Lucille A. Cola	662-519-0717	785 The Corner
7	Paul J. Clark	484-325-1544	125Good View

Primary Key

Table: Member

Key and Relationship

Presentation of Primary Key

Photo		
	ID	integer(5)
	Title	varchar(120)
	Description	varchar(255)
	Privacy	varchar(20)
	UploadDat	date
	View	integer(10)

Key and Relationship

Picking a Primary Key

Check the following when you evaluate candidates for a table's primary key

Unique?

Mandatory?

Constant?

Key and Relationship

Picking a Primary Key

Member Name

Unique?

Mandatory?

Constant?

Key and Relationship

Picking a Primary Key

Member's SSN

Unique?

Mandatory?

Constant?

Key and Relationship

Picking a Primary Key

Member Email Address

Unique?

Mandatory?

Constant?

Key and Relationship

Picking a Primary Key

Computer generated ID

Unique?

Mandatory?

Constant?

Key and Relationship

Primary key could be a single column or
a group of columns.

Key and Relationship

When you define more than one column as a primary key, it's called a **composite primary key**.

Key and Relationship

Composite PK Example



Photo ID	Member ID	Shared Date
1	1	3/3/2015
1	2	3/4/2015
1	3	3/5/2015
2	1	3/6/2015
2	4	3/7/2015
3	1	3/2/2015
3	5	3/3/2015

Table: Photo_Sharing

Key and Relationship

Composite PK Example

Not unique

Not unique

Photo ID	Member ID	Shared Date
1	1	3/3/2015
1	2	3/4/2015
1	3	3/5/2015
2	1	3/6/2015
2	2	3/7/2015
3	1	3/2/2015
3	5	3/3/2015

Table: Photo_Sharing

Key and Relationship

Composite PK Example

Unique

Photo ID	Member ID	Shared Date
1	1	3/3/2015
1	2	3/4/2015
1	3	3/5/2015
2	1	3/6/2015
2	4	3/7/2015
3	1	3/2/2015
3	5	3/3/2015

Table: Photo_Sharing

Key and Relationship

Composite PK Example

Photo_Sharing

- 🔑 Photo ID : integer (10)
- 🔑 Member ID : integer (10)
- Shared Date: date

Photo ID	Member ID	Shared Date
1	1	3/3/2015
1	2	3/4/2015
1	3	3/5/2015
2	1	3/6/2015
2	4	3/7/2015
3	1	3/2/2015
3	5	3/3/2015

Table: Photo_Sharing

Key and Relationship

Primary Key Assignment

Values of a primary key can be assigned by the following ways



Key and Relationship

Primary Key Assignment

Auto-Generated

- Allows a unique value to be generated when you insert a new record into a table.

ID
1
2
3
+ 4

Key and Relationship

Primary Key Assignment

Specify Manually

- Provide a unique value when you insert a new record into a table.
- The value is mandatory (cannot be unfilled/null)
- Examples:
 - Social Security Number (SSN)
 - Email address
 - ISBN

SSN
751-15-2261
122-21-7421
693-86-0582
+ 830-27-5782

Key and Relationship

Primary Key Assignment

Sequence

- Allows a unique value to be generated when you insert a new record into a table.
- The **interval**, **min value**, **max value** and **sequence** of value can be defined.
- Example:

ID
5
10
15
+ 20

Key and Relationship

Ascending Sequence

MINVALUE: 1

MAXVALUE: 99999999

START WITH: 1

INCREMENT BY: 3

CACHE: 20

The number of sequence values to be stored in memory for faster access.

Ascending Sequence

MINVALUE: 1
MAXVALUE: 99999999
START WITH: 1
INCREMENT BY: 3
CACHE: 20

ID
1
4
7
10

Key and Relationship

Descending Sequence

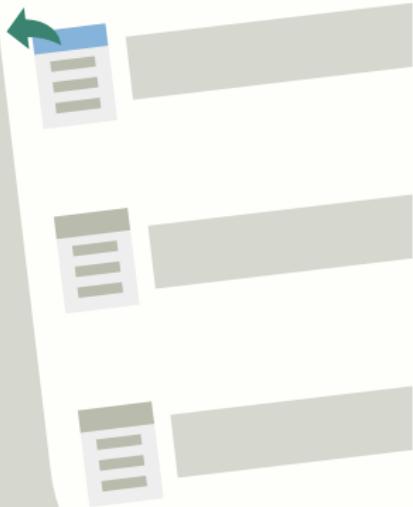
MINVALUE: 1
MAXVALUE: 50000000
START WITH: 50000000
INCREMENT BY: -10
NOCACHE

Descending Sequence

MINVALUE: 1
MAXVALUE: 50000000
START WITH: 50000000
INCREMENT BY: -10
NOCACHE

ID
50000000
49999990
49999980
49999970
⋮

Key and Relationship



Foreign Key

Key and Relationship

Foreign key is a table column which
uniquely identifies a record in another table.

Key and Relationship

Understanding Foreign Key

Photo Details

ID	Title	Member Name	Member PhoneNum	Member Address
1	The tram	Chris Wong	252-608-5628	572 Lost Lake Crossing
2	Little kitten	Chris Wong	252-608-5628	572 Lost Lake Crossing
3	Chamomile	Chris Wong	252-608-5628	572 Lost Lake Crossing
4	New bus route	Wanda A. Allison	808-839-2217	2017 Green Impasse
5	Ferry pier 9	Chris Wong	252-608-5628	572 Lost Lake Crossing
6	The coast	Chris Wong	252-608-5628	572 Lost Lake Crossing
7	Rock climbing	Wanda A. Allison	808-839-2217	2017 Green Impasse
8	Fountain	Chris Wong	252-608-5628	572 Lost Lake Crossing
9	Sunset	Wanda A. Allison	808-839-2217	2017 Green Impasse

Database table: Photo

Key and Relationship

Understanding Foreign Key

Member Details

ID	Title	Member Name	Member PhoneNum	Member Address
1	The tram	Chris Wong	252-608-5628	572 Lost Lake Crossing
2	Little kitten	Chris Wong	252-608-5628	572 Lost Lake Crossing
3	Chamomile	Chris Wong	252-608-5628	572 Lost Lake Crossing
4	New bus route	Wanda A. Allison	808-839-2217	2017 Green Impasse
5	Ferry pier 9	Chris Wong	252-608-5628	572 Lost Lake Crossing
6	The coast	Chris Wong	252-608-5628	572 Lost Lake Crossing
7	Rock climbing	Wanda A. Allison	808-839-2217	2017 Green Impasse
8	Fountain	Chris Wong	252-608-5628	572 Lost Lake Crossing
9	Sunset	Wanda A. Allison	808-839-2217	2017 Green Impasse

Database table: Photo

Key and Relationship

Understanding Foreign Key

Data kept repeating

ID	Title	Member Name	Member PhoneNum	Member Address
1	The tram	Chris Wong	252-608-5628	572 Lost Lake Crossing
2	Little kitten	Chris Wong	252-608-5628	572 Lost Lake Crossing
3	Chamomile	Chris Wong	252-608-5628	572 Lost Lake Crossing
4	New bus route	Wanda A. Allison	808-839-2217	2017 Green Impasse
5	Ferry pier 9	Chris Wong	252-608-5628	572 Lost Lake Crossing
6	The coast	Chris Wong	252-608-5628	572 Lost Lake Crossing
7	Rock climbing	Wanda A. Allison	808-839-2217	2017 Green Impasse
8	Fountain	Chris Wong	252-608-5628	572 Lost Lake Crossing
9	Sunset	Wanda A. Allison	808-839-2217	2017 Green Impasse

Database table: Photo

Key and Relationship

Understanding Foreign Key

ID	Title	Member ID
1	The tram	1
2	Little kitten	1
3	Chamomile	1
4	New bus route	2
5	Ferry pier 9	1
6	The coast	1
7	Rock climbing	2
8	Fountain	1
9	Sunset	2

Database table: **Photo**

ID	Name	PhoneNum	Address
1	Chris Wong	252-608-5628	572 Lost Lake Crossing
2	Wanda A. Allison	808-839-2217	2017 Green Impasse

Database table: **Member**



Photo and Member details
in **separate tables**

Key and Relationship

Understanding Foreign Key

ID	Title	Member ID
1	The tram	1
2	Little kitten	1
3	Chamomile	1
4	New bus route	2
5	Ferry pier 9	1
6	The coast	1
7	Rock climbing	2
8	Fountain	1
9	Sunset	2

Database table: Photo

ID	Name	PhoneNum	Address
1	Chris Wong	252-608-5628	572 Lost Lake Crossing
2	Wanda A. Allison	808-839-2217	2017 Green Impasse

Database table: Member

Foreign Key Column

Key and Relationship

Understanding Foreign Key

The diagram illustrates a foreign key relationship between two database tables: **Photo** and **Member**.

Database table: Photo

ID	Title	Member ID
1	The tram	1
2	Little kitten	1
3	Chamomile	1
4	New bus route	2
5	Ferry pier 9	1
6	The coast	1
7	Rock climbing	2
8	Fountain	1
9	Sunset	2

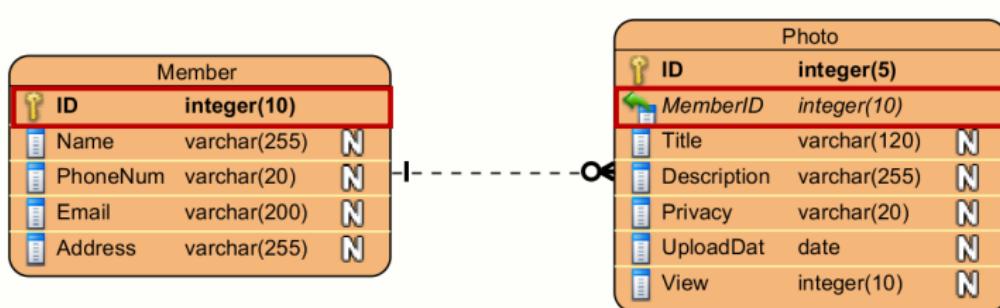
Database table: Member

ID	Name	PhoneNum	Address
1	Chris Wong	252-608-5628	572 Lost Lake Crossing
2	Wanda A. Allison	808-839-2217	2017 Green Impasse

A red callout bubble points to the value '1' in the **Member ID** column of the **Photo** table, stating: "Uniquely identifies a member in the Member table".

Key and Relationship

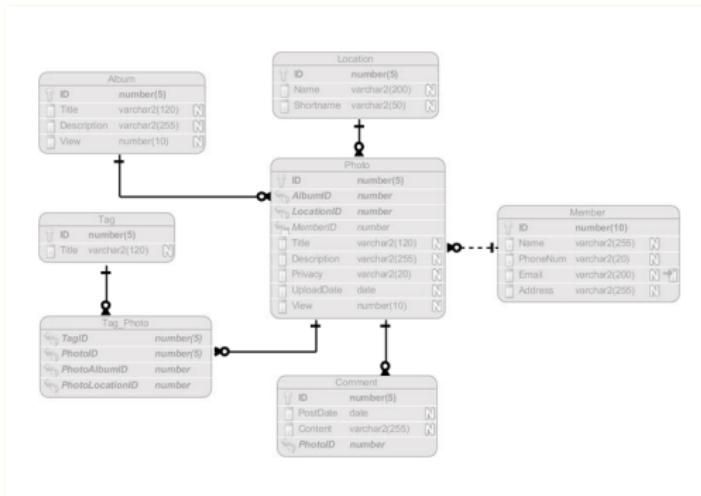
Presentation of Foreign Key



The use of foreign key in Photo entity (Referencing the **primary key** of Member entity)

Key and Relationship

Relationship and cardinality



Key and Relationship

Relationship and cardinality

Common cardinality types:

1 to many

1 to 0..1

Many to Many

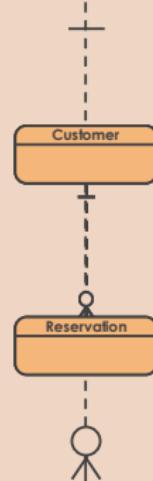
Key and Relationship



Relationship and cardinality

■ 1 to Many

- Used when a record in a table can reference multiple records in another table.
 - Examples
 - A **Member** can upload many **Photos**
 - A **Customer** can make many **Reservations**
 - A **Document** can have many **Revisions**
- Probably the most common type of cardinality

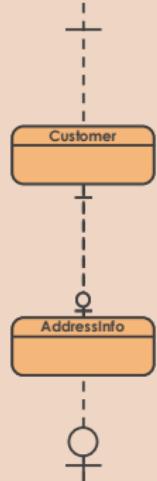


Key and Relationship

Relationship and cardinality

■ 1 to 0..1

- Used when a record in a table can reference up to one record in another table.
 - Examples
 - A **BlogPost** can have a single piece of blog **Content**
 - A **Customer** can have one set of **AddressInfo**
- You gain performance due to:
 - Separating less commonly used data into another table.
- You lose performance due to:
 - More tables lead to more table overhead and disc storage.
 - Complicated queries resulted by additional JOIN clause.
- Conclusion:
 - If there are data that you don't access in >90% of your use cases (or you don't have that data in most of the time), then it would be a wise choice to separate these data into another table for better performance.

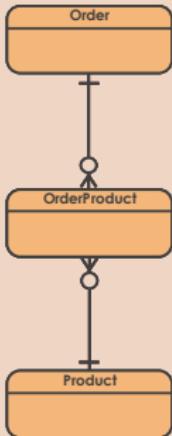


Key and Relationship

Relationship and cardinality

Many to Many

- Broken down into a pair of one-to-many relationships.
- Associative entity is used to define the association between two related entities
- Used when a record in a parent table can reference multiple records in another table. At the same time, a record in a child table can also reference multiple records in the parent table.
 - Examples
 - A **Student** can enroll in many **Courses**, while a **Course** can be taken by many **Students**
 - An **Order** can consist of many **Products**, while a **Product** can appear in many **Orders**.



Key and Relationship

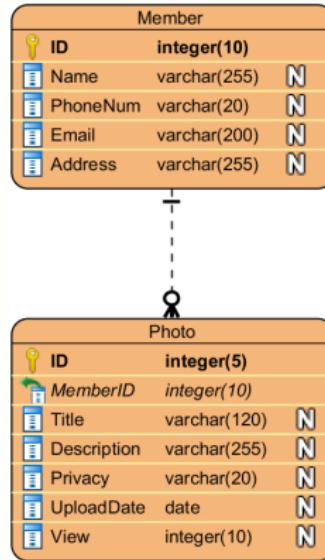
Foreign Key Example



A Member
can upload
many Photos

Key and Relationship

Foreign Key Example



Key and Relationship

Foreign Key Example



A **BlogPost** can have a single piece of blog Content

Key and Relationship

Foreign Key Example

This is a blog post

Aliquam erat volutpat. Ut semper enim eu nibh tempus condimentum. Quisque suscipit. Ut eu ligula ac nunc suscipit dictum. Vestibulum purus ante, auctor vel, viverra quis, nonummy et, pede. Curabitur lacus mi, iaculis eget, luctus et, consequat quis, lacus. Morbi pellentesque tortor eget diam. Nunc blandit nunc vel eros. Duis venenatis. Ut consequat ultrices erat. Nullam eget nulla. Sed bibendum. Cras semper ipsum ut neque faucibus feugiat.

BlogPost		
ID	integer(10)	N
Title	varchar(100)	N
Summary	varchar(100)	N
PostDate	date	N
Status	char(5)	N
ShortName	varchar(20)	N

Content		
ID	integer(10)	
BlogPostID	integer(10)	
content	clob	N

Key and Relationship

Foreign Key Example



hongkong → pier → ferry pier
ferry → central → building
nine →

A **Photo** can have many **Tags**.

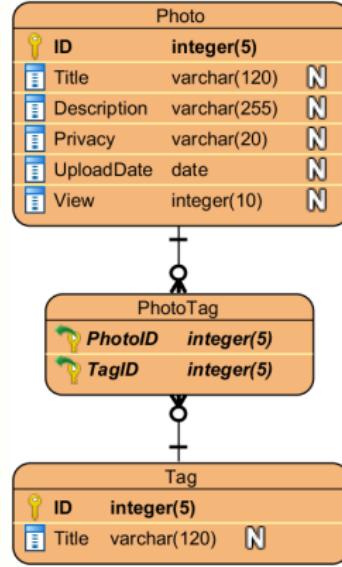
A **Tag** can be used by many **Photos**.

Key and Relationship

Foreign Key Example

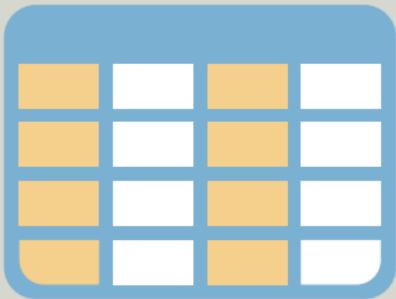


hongkong → pier → ferry pier
ferry → central → building
nine →



Key and Relationship

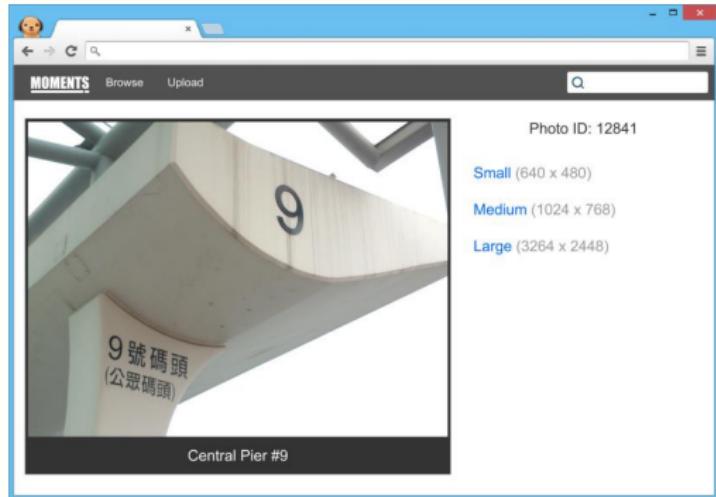
Plan



Database view is a virtual table that provides the result set of a pre-established query command.

Database View

Concept of DB View



Database View

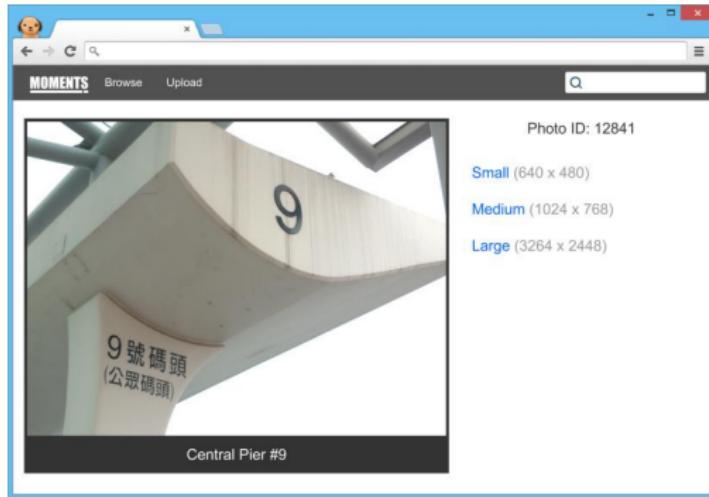
Concept of DB View

Photo

ID : number (5)
Title : varchar (120)
Description : varchar (255)
Privacy : varchar (20)
UploadDate : date
View : number (10)
ImagePath: varchar (50)



Source of data



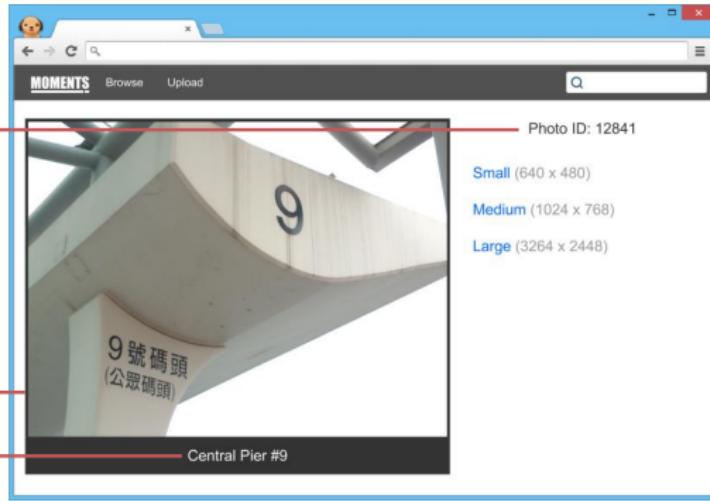
Database View

Concept of DB View

Just some columns are queried



Photo	
ID : number	(5)
Title : varchar	(120)
Description : varchar	(255)
Privacy : varchar	(20)
UploadDate : date	
View : number	(10)
ImagePath: varchar	(50)



Database View

Concept of DB View

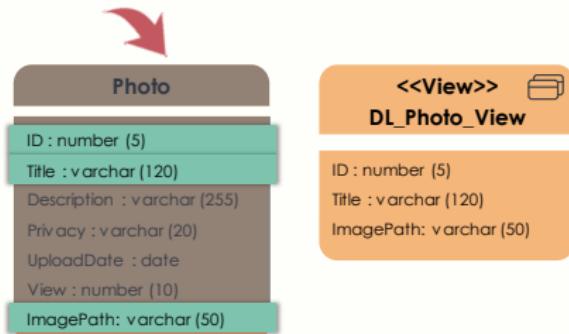
A database view specialized for querying
the data required by the download page



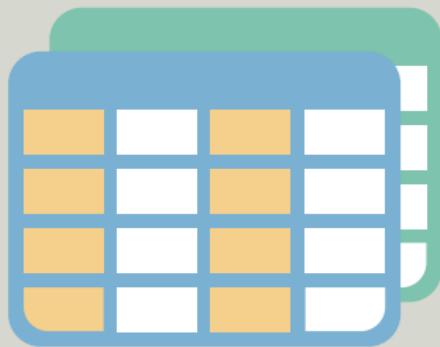
Database View

Concept of DB View

Add the columns needed by the view



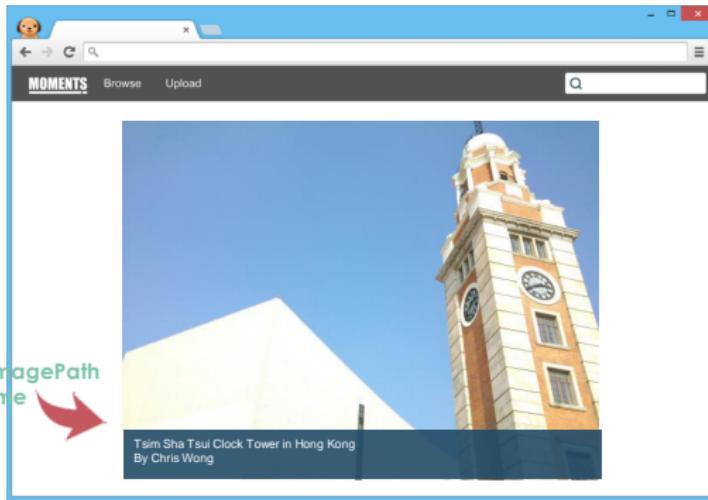
Database View



You can grab data from
multiple tables by using **join**

Database View

Concept of Joining Tables

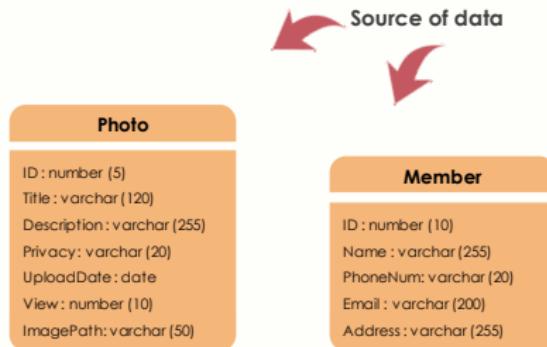


Source of data:

Photo table: Title, ImagePath
Member table: Name

Database View

Concept of Joining Tables



Database View

Concept of Joining Tables

Photo
ID : number (5)
Title : varchar (120)
Description : varchar (255)
Privacy : varchar (20)
UploadDate : date
View : number (10)
ImagePath: varchar (50)

Member
ID : number (10)
Name : varchar (255)
PhoneNum: varchar (20)
Email : varchar (200)
Address : varchar (255)

<<View>> 
Browse_Photo_View

A database view for browse photo page



Database View

Concept of Joining Tables

Tables can be joined

Photo
ID : number (5)
Title : varchar (120)
Description : varchar (255)
Privacy : varchar (20)
UploadDate : date
View : number (10)
ImagePath: varchar (50)

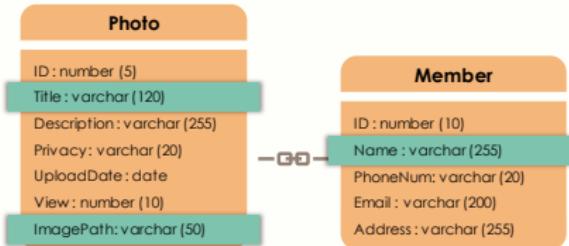
—◎—

Member
ID : number (10)
Name : varchar (255)
PhoneNum: varchar (20)
Email : varchar (200)
Address : varchar (255)

<<View>> 
Browse_Photo_View

Database View

Concept of Joining Tables



->-

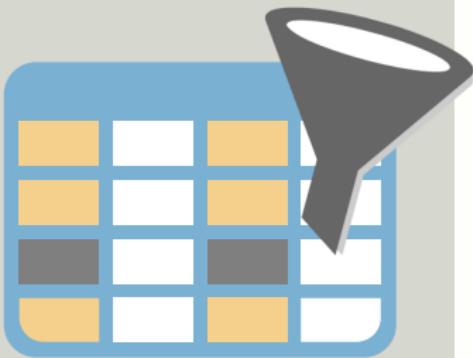
<<View>> 
Browse_Photo_View

A view with columns from
multiple tables



PhotoTitle : varchar (120)
ImagePath : varchar (50)
MemberName: varchar (255)

Database View

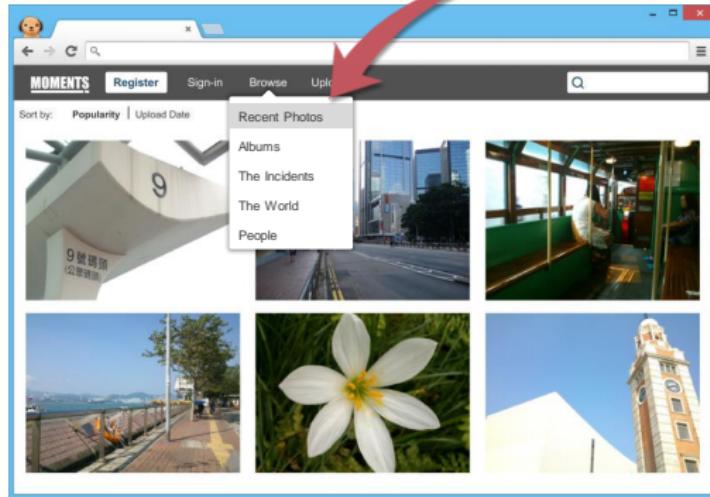


You can apply **Filter** to ensure
that only relevant data is
stored in the result set

Database View

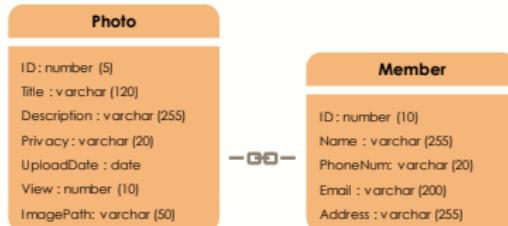
Concept of Filter

How about a view that lists
only recent photos' data?



Database View

Concept of Filter



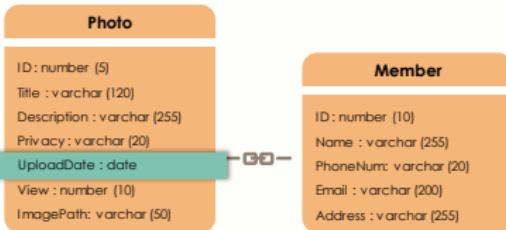
A view that contains both photo and member data.

Database View

Concept of Filter

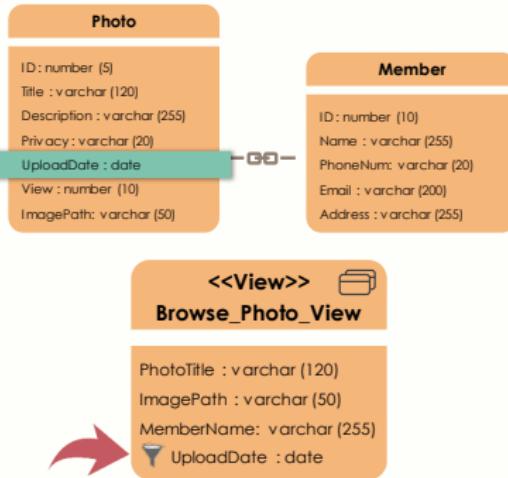


To list out only the recent photos, we need the help of the **UploadDate** column.



Database View

Concept of Filter



Apply a filter to the column

Database View

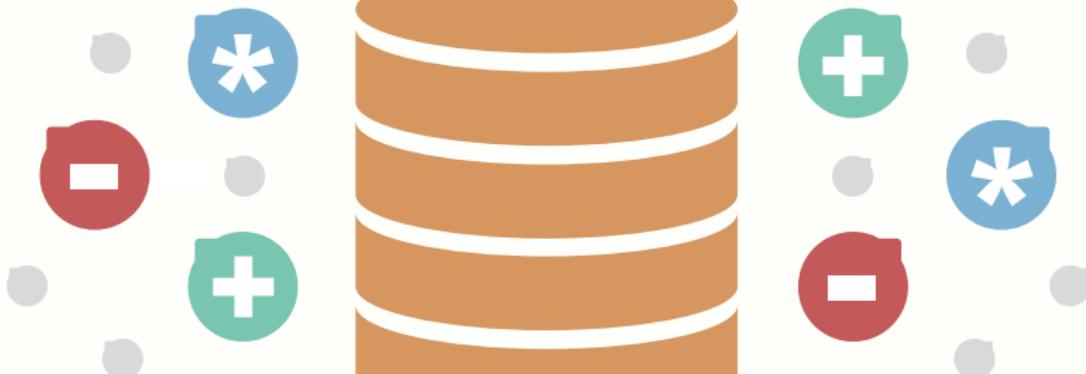
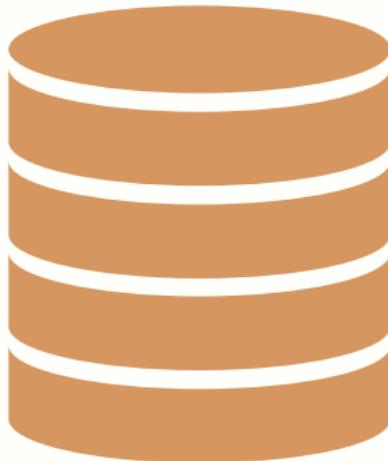
Plan



A database trigger is a procedure that is automatically executed in response to certain events on a database table

Triggers and Stored Procedures

Sample Usage - Auditing



As the system runs, you keep modify the data in database...

Triggers and Stored Procedures

Sample Usage - Auditing



Who modified the price
of milk to \$320?

When did he/she do it?

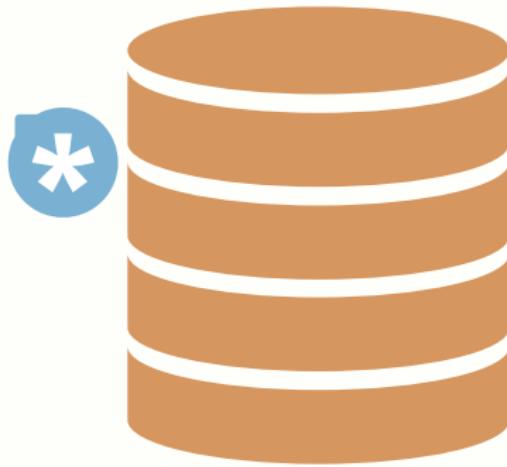
What was the value before the
change?
...

Product	Price
Biscuit	10
Bread	8
Milk	320
Sugar	6

For business reasons, you may want to know the answer of these questions.

Triggers and Stored Procedures

Sample Usage - Auditing



Trigger can be used to audit your database. Let's say if someone is trying to update a record

Triggers and Stored Procedures

Sample Usage - Auditing



Type	UserName	Time	Old_Val	New_Val
U	WG\Peter	2/5/...	8	320

If you have written a trigger for auditing, it can help you log his action before writing data into DB

Triggers and Stored Procedures



A stored procedure is a pre-written procedure code that allows you to execute over and over again for validation or quick retrieval of data.

Triggers and Stored Procedures

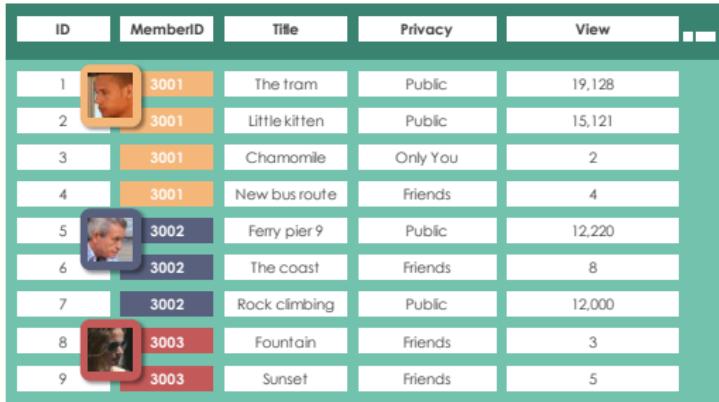
Sample Usage – Querying Data

ID	MemberID	Title	Privacy	View
1	3001	The tram	Public	19,128
2	3001	Little kitten	Public	15,121
3	3001	Chamomile	Only You	2
4	3001	New bus route	Friends	4
5	3002	Ferry pier 9	Public	12,220
6	3002	The coast	Friends	8
7	3002	Rock climbing	Public	12,000
8	3003	Fountain	Friends	3
9	3003	Sunset	Friends	5

Let's take a look at this table. This is the **Photo** table, with all photos listed.

Triggers and Stored Procedures

Sample Usage – Querying Data



ID	MemberID	Title	Privacy	View
1	3001	The tram	Public	19,128
2	3001	Little kitten	Public	15,121
3	3001	Chamomile	Only You	2
4	3001	New bus route	Friends	4
5	3002	Ferry pier 9	Public	12,220
6	3002	The coast	Friends	8
7	3002	Rock climbing	Public	12,000
8	3003	Fountain	Friends	3
9	3003	Sunset	Friends	5

These photos are uploaded by different members, identified by their IDs.

Triggers and Stored Procedures

Sample Usage – Querying Data



ID	MemberID	Title	Privacy	View
1	3001	The tram	Public	19,128
2	3001	Little kitten	Public	15,121
3	3001	Chamomile	Only You	2
4	3001	New bus route	Friends	4
5	3002	Ferry pier 9	Public	12,220
6	3002	The coast	Friends	8
7	3002	Rock climbing	Public	12,000
8	3003	Fountain	Friends	3
9	3003	Sunset	Friends	5

The photos have different levels of privacy – public, only you (private) and friends.

Triggers and Stored Procedures

Sample Usage – Querying Data



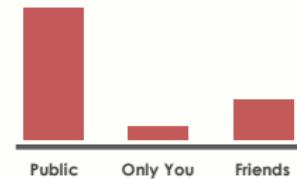
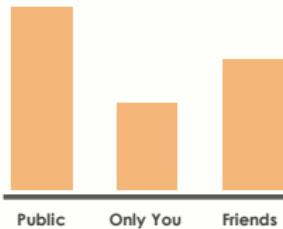
ID	MemberID	Title	Privacy	View
1	3001	The tram	Public	19,128
2	3001	Little kitten	Public	15,121
3	3001	Chamomile	Only You	2
4	3001	New bus route	Friends	4
5	3002	Ferry pier 9	Public	12,220
6	3002	The coast	Friends	8
7	3002	Rock climbing	Public	12,000
8	3003	Fountain	Friends	3
9	3003	Sunset	Friends	5

and the number of view for each photo is stored in the View column.

Triggers and Stored Procedures

Sample Usage – Querying Data

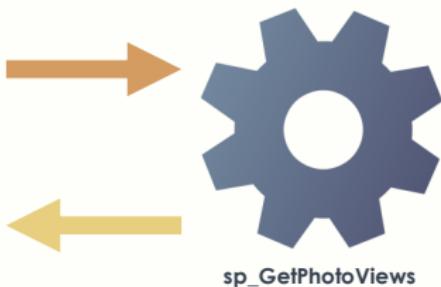
Photo view count



If now you want to retrieve the photo view count of different members, categorized by the privacy of photo...

Triggers and Stored Procedures

Sample Usage – Querying Data



MemberID	Public_Count	Only_You_Count	Friend_Count
3001	34249	2	4
3002	24220	0	8
3003	0	0	8
⋮			

You can write a procedure to return the resultset you need. By executing the procedure you will get the data you need easily.

Triggers and Stored Procedures

Creating a Procedure

```
CREATE PROCEDURE sp_GetPhotoViews()
BEGIN
    DECLARE v_member_id int;
    DECLARE v_pub_photo_viewcount int DEFAULT 0;
    DECLARE v_pri_photo_viewcount int DEFAULT 0;
    DECLARE v_frd_photo_viewcount int DEFAULT 0;
    DECLARE v_member_cursor CURSOR FOR SELECT ID, 'Name' FROM Member;

    CREATE TEMPORARY TABLE member_photo_viewcount (member_id int, public_photo_viewcount int, private_photo_viewcount int, friends_photo_viewcount int);

    OPEN v_member_cursor;

    FETCH v_member_cursor into v_member_id;
    BEGIN
        SELECT SUM('View') INTO v_pub_photo_viewcount FROM Photo WHERE Privacy = 'Public' AND MemberID = v_member_id;
        SELECT SUM('View') INTO v_pri_photo_viewcount FROM Photo WHERE Privacy = 'Only You' AND MemberID = v_member_id;
        SELECT SUM('View') INTO v_frd_photo_viewcount FROM Photo WHERE Privacy = 'Friends' AND MemberID = v_member_id;

        INSERT INTO member_photo_viewcount VALUES (v_member_id, v_pub_photo_viewcount, v_pri_photo_viewcount, v_frd_photo_viewcount);
        FETCH v_member_cursor into v_member_id;
    END;

    CLOSE v_member_cursor;

    SELECT * FROM member_photo_viewcount;
    DROP TABLE member_photo_viewcount;
END
```

Triggers and Stored Procedures

End of Lecture 4.

TDT5FTOTTC



Top 5 Fundamental Takeaways

Top 5 Fundamental Takeaways

5 Database Views, Triggers, and Stored Procedures

enhance database security, automation, and efficiency by providing pre-queried data, event-based actions, and reusable SQL routines.

Top 5 Fundamental Takeaways

- 5 Database Views, Triggers, and Stored Procedures** enhance database security, automation, and efficiency by providing pre-queried data, event-based actions, and reusable SQL routines.
- 4 Keys and Relationships** define how data is uniquely identified (Primary Keys), linked across tables (Foreign Keys), and structured using Composite Keys for complex relationships.

Top 5 Fundamental Takeaways

- 5 **Database Views, Triggers, and Stored Procedures** enhance database security, automation, and efficiency by providing pre-queried data, event-based actions, and reusable SQL routines.
- 4 **Keys and Relationships** define how data is uniquely identified (Primary Keys), linked across tables (Foreign Keys), and structured using Composite Keys for complex relationships.
- 3 **Normalization** and Database Optimization help minimize data redundancy and balance between normalized (integrity-focused) and denormalized (performance-oriented) structures.

Top 5 Fundamental Takeaways

- 5 **Database Views, Triggers, and Stored Procedures** enhance database security, automation, and efficiency by providing pre-queried data, event-based actions, and reusable SQL routines.
- 4 **Keys and Relationships** define how data is uniquely identified (Primary Keys), linked across tables (Foreign Keys), and structured using Composite Keys for complex relationships.
- 3 **Normalization** and Database Optimization help minimize data redundancy and balance between normalized (integrity-focused) and denormalized (performance-oriented) structures.
- 2 **Three Levels of ER Modeling** include the conceptual (high-level overview), logical (detailed structure with normalization), and physical (implementation-specific design).

Top 5 Fundamental Takeaways

- 5 **Database Views, Triggers, and Stored Procedures** enhance database security, automation, and efficiency by providing pre-queried data, event-based actions, and reusable SQL routines.
- 4 **Keys and Relationships** define how data is uniquely identified (Primary Keys), linked across tables (Foreign Keys), and structured using Composite Keys for complex relationships.
- 3 **Normalization** and Database Optimization help minimize data redundancy and balance between normalized (integrity-focused) and denormalized (performance-oriented) structures.
- 2 **Three Levels of ER Modeling** include the conceptual (high-level overview), logical (detailed structure with normalization), and physical (implementation-specific design).
- 1 **ERDs (Entity-Relationship Diagrams)** provide a visual representation of database structures and relationships to improve data organization and communication.

Top 5 Fundamental Takeaways

- 5 **Database Views, Triggers, and Stored Procedures** enhance database security, automation, and efficiency by providing pre-queried data, event-based actions, and reusable SQL routines.
- 4 **Keys and Relationships** define how data is uniquely identified (Primary Keys), linked across tables (Foreign Keys), and structured using Composite Keys for complex relationships.
- 3 **Normalization** and Database Optimization help minimize data redundancy and balance between normalized (integrity-focused) and denormalized (performance-oriented) structures.
- 2 **Three Levels of ER Modeling** include the conceptual (high-level overview), logical (detailed structure with normalization), and physical (implementation-specific design).
- 1 **ERDs (Entity-Relationship Diagrams)** provide a visual representation of database structures and relationships to improve data organization and communication.

Database Administration: Reverse Design.

Udemy Explorar Udemy Business Instructor Mi aprendizaje

Desarrollo > Diseño y desarrollo de bases de datos > Sistemas de gestión de bases de datos (DBMS)



Vista previa de este curso

Database Design and Management

Learn how to design and manage database with ERD, database generation and reversal with Visual Paradigm.

Tutorial gratuito **4.5** (3 458 calificaciones) 95.409 estudiantes
12 h 41 min de video bajo demanda

Created by [Visual Paradigm](#)
Inglés Inglés [automático]

[Ir al curso](#)

Lo que aprenderás [Contenido del curso](#) Valoraciones Instructores

Requisitos

- Basic software development concept
- PC or OSX or Linux

Contenido del curso

4 secciones • 14 clases • 2 h 41 m de duración total [Ampliar todas las secciones](#)

▼ Introduction to Database Design and Management	8 clases • 1 h 31 min
▼ Database Engineering	3 clases • 37 min
▼ Java Database Programming	1 clases • 17 min
▼ Best Practices	2 clases • 17 min

Content has been extracted from *Database Design and Management*. Udemy Course, created by Visual Paradigm, 2025. Visit <https://www.udemy.com/course/database-design-and-management/> and <https://www.visual-paradigm.com/>.