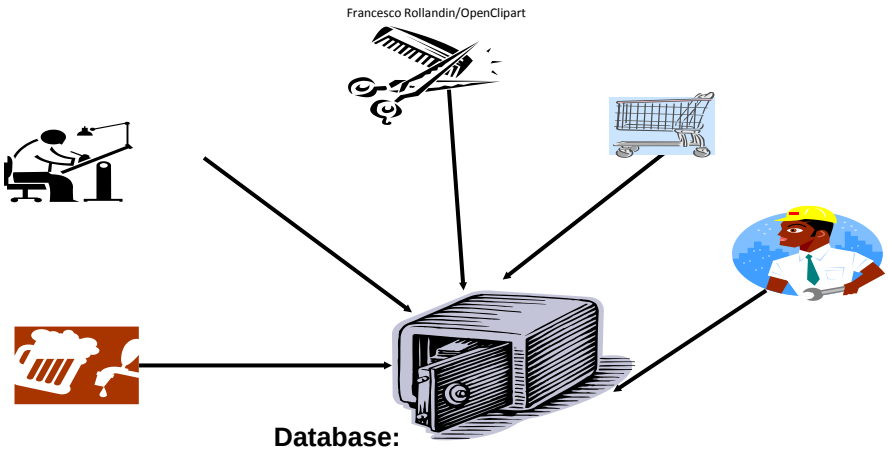
Databases

In this section you will learn about: how information is stored in databases, the different types of relations that can exist within a database, and how information can be retrieved via queries.

Online MS-Office information source:
https://support.office.com/

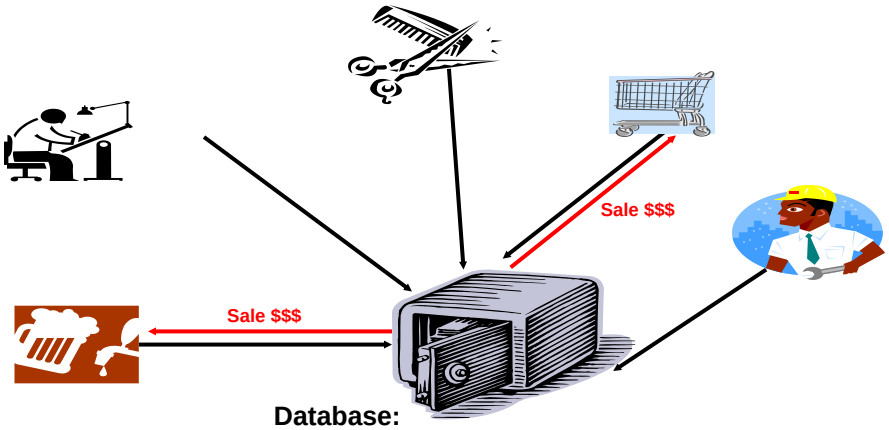
Purpose Of A Database

- To store information



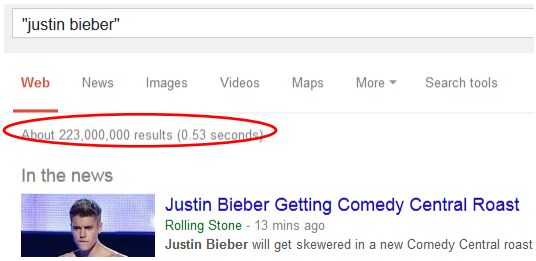
Purpose Of A Database

- To retrieve information information



Databases: Storing / Retrieving Information

- As you will see, implementing these two tasks aren't as easy as it seems.
- Information must be stored such that:
  - Information can be quickly retrieved



Databases: Storing / Retrieving Information (2)

– The database is designed to reduce problems during maintenance (additions, modifications, deletions)
  • Example: You may see this issue arise if talk about database normalization.

Marketing Dept.
• Loren Coleman
• William McCloud

One employee has left and the whole department is gone?

Finance & Accounting
• Victor Davion
• Ester Flowers

Databases: Storing / Retrieving Information (3)

– Minimizes redundancy:

Students data base table

ID	First Name	Last Name	Phone	Class 1	Class 2
123456	Jamie	Smyth	553-3992	CPSC 203, 01	PSYC 205, 03
123457	Stacey	Walls	790-3992	ACCT 321, 02	FNCE 353, 05
123458	Angel	Lam	551-4993	MATH 211, 02	MATH 251, 01

Classes data base table

ClassName	*ClassNumber*	*Lecture No*	ClassDescription
CPSC	*203*	*01*	Introduction to Computers
CPSC	*231*	*01*	Introduction to Computer Science I
CPSC	*233*	*01*	Introduction to Computer Science II

With Bother With Databases?

- Are used to store and retrieve information
- Why bother, use a simple file as an alternative?
  - E.g., tracking client information

MILES EDWARD O'BRIAN
DS9 Corp
Electrical engineering
2007 purchases: $10,0000,000
2006 purchases: $1,750,000

JAMIE SMYTHE
Cooperative services
Gasoline refining
2006 purchases: $5,000,0000
2005 purchases: $5,000,0000
2004 purchases: $5,000,0000
2003 purchases: $5,000,0000
2002 purchases: $5,000,0000

SCOTT BRUCE
Bryce Consulting
Investment analysis
2007 purchases: $500,000
2006 purchases: $1,500,000
2005 purchases: $2,500,000
2004 purchases: $500,000

Etc.

- If the list is short then a simple text file may suffice
- As the list grows organizing and updating the information becomes more challenging (duplicates or inaccuracies?)
- Also searching the list according to specific criteria may become difficult
  - e.g., Show all clients whose purchases in 2007 were between one and five million dollars
  - e.g., Show all clients that made a purchase exceeding 10 million dollars.

Storing Information In A Database

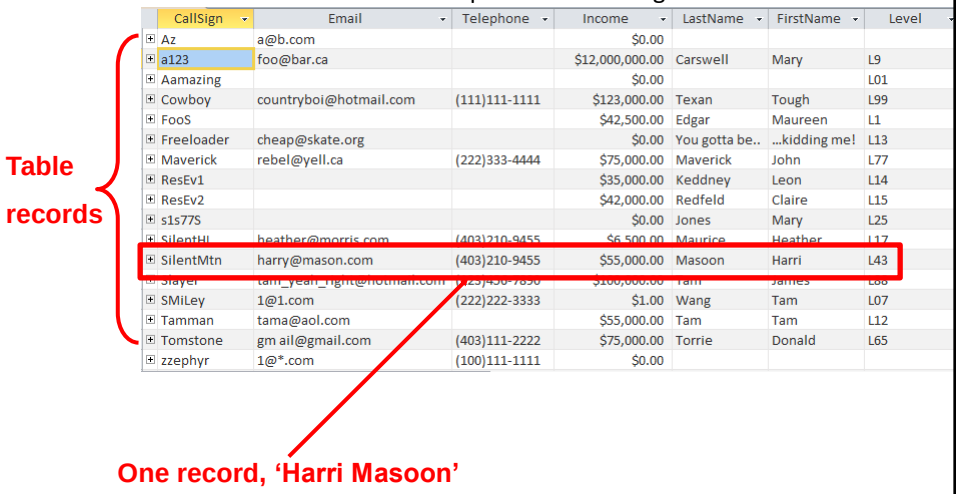
- Information is stored in tables:

The 'Gamers' table

CallSign	Email	Telephone	Income	LastName	FirstName	Level
Az	a@b.com		$0.00			
a123	foo@bar.ca		$12,000,000.00	Carswell	Mary	L9
Aamazing			$0.00			L01
Cowboy	countryboi@hotmail.com	(111)111-1111	$123,000.00	Texan	Tough	L99
FooS			$42,500.00	Edgar	Maureen	L1
Freeloader	cheap@skate.org		$0.00	You gotta be..	...kidding me!	L13
Maverick	rebel@yell.ca	(222)333-4444	$75,000.00	Maverick	John	L77
ResEv1			$35,000.00	Keddney	Leon	L14
ResEv2			$42,000.00	Redfeld	Claire	L15
s1s77S			$0.00	Jones	Mary	L25
SilentHL	heather@morris.com	(403)210-9455	$6,500.00	Maurice	Heather	L17
SilentMtn	harry@mason.com	(403)210-9455	$55,000.00	Masoon	Harri	L43
Slayer	tam_yeah_right@hotmail.com	(123)456-7890	$100,000.00	Tam	James	L88
SMiLey	1@1.com	(222)222-3333	$1.00	Wang	Tam	L07
Tamman	tama@aol.com		$55,000.00	Tam	Tam	L12
Tomstone	gm ail@gmail.com	(403)111-2222	$75,000.00	Torrie	Donald	L65
zzephyr	1@*.com	(100)111-1111	$0.00			

Storing Information In A Database (2)

- Row = Record: An example instance of data within the table.
  - Gamers Table: one row is an example instance of a gamer



CallSign	Email	Telephone	Income	LastName	FirstName	Level
Az	a@b.com		$0.00			
a123	foo@bar.ca		$12,000,000.00	Carswell	Mary	L9
Aamazing			$0.00			L01
Cowboy	countryboi@hotmail.com	(111)111-1111	$123,000.00	Texan	Tough	L99
FooS			$42,500.00	Edgar	Maureen	L1
Freeloader	cheap@skate.org		$0.00	You gotta be..	...kidding me!	L13
Maverick	rebel@yell.ca	(222)333-4444	$75,000.00	Maverick	John	L77
ResEv1			$35,000.00	Keddney	Leon	L14
ResEv2			$42,000.00	Redfeld	Claire	L15
s1s77S			$0.00	Jones	Mary	L25
SilentHL	heather@morris.com	(403)210-9455	$6,500.00	Maurice	Heather	L17
SilentMtn	harry@mason.com	(403)210-9455	$55,000.00	Masoon	Harri	L43
Slayer	tam_yeah_right@hotmail.com	(123)456-7890	$100,000.00	Tam	James	L88
SMiLey	1@1.com	(222)222-3333	$1.00	Wang	Tam	L07
Tamman	tama@aol.com		$55,000.00	Tam	Tam	L12
Tomstone	gm ail@gmail.com	(403)111-2222	$75,000.00	Torrie	Donald	L65
zzephyr	1@*.com	(100)111-1111	$0.00			

Storing Information In A Database (3)

- Column: are that attributes that we track for each record
  - Gamers Table: each column specifies the information we store about the games in this database.

Attributes ('fields' in Access) of each record

CallSign	Email	Telephone	Income	LastName	FirstName	Level
Az	a@b.com		$0.00			
a123	foo@bar.ca		$12,000,000.00	Carswell	Mary	L9
Aamazing			$0.00			L01
Cowboy	countryboi@hotmail.com	(111)111-1111	$123,000.00	Texan	Tough	L99
FooS			$42,500.00	Edgar	Maureen	L1
Freeloader	cheap@skate.org		$0.00	You gotta be..	...kidding me!	L13
Maverick	rebel@yell.ca	(222)333-4444	$75,000.00	Maverick	John	L77
ResEv1			$35,000.00	Keddney	Leon	L14
ResEv2			$42,000.00	Redfeld	Claire	L15
s1s77S			$0.00	Jones	Mary	L25
SilentHL	heather@morris.com	(403)210-9455	$6,500.00	Maurice	Heather	L17
SilentMtn	harry@mason.com	(403)210-9455	$55,000.00	Masoon	Harri	L43
Slayer	tam_yeah_right@hotmail.com	(123)456-7890	$100,000.00	Tam	James	L88
SMiLey	1@1.com	(222)222-3333	$1.00	Wang	Tam	L07
Tamman	tama@aol.com		$55,000.00	Tam	Tam	L12
Tomstone	gm ail@gmail.com	(403)111-2222	$75,000.00	Torrie	Donald	L65
zzephyr	1@*.com	(100)111-1111	$0.00			

Primary Key

- Each table should typically have one attribute designated as the primary key:
  - The primary key must be guaranteed to be unique
  - It must uniquely identify one record from another

SIN	LastName	FirstName	Address	City	Province
638666670	Cartland	Douglas	1109, 4944 Dalworth Dr	Silent Hill	Alberta
456789123	Cartman	Eric	456 Lynchview Road	Southpark	Alberta
670380456	Edgar	Maureen	300, Lockinvar Road	Calgary	Alberta
456889123	Flanders	Ned	60 Evergreen Terrace	Springfield	Alberta
413754621	Kennedy	Leon	808, 4900 Wildman Ave	Racoon City	Alberta
456438624	Lemoy	Leonard	55 Logic Way	Vulcan	Alberta
666666667	Mason	Harry	7 Luckstone Dr	Silent Hill	Alberta
666666666	Morris	Heather	7 Luckstone Dr	Silent Hill	Alberta
444638047	Redfield	Claire	653 Wildpark Place	Racoon City	Alberta
123115323	Simcox	Cole	311 Ocean View Drive	Vancouver	British C
456789124	Simpson	Homer	59 Evergreen Terrace	Springfield	Alberta
123456789	Smith	John	123 Peanut Lane	Calgary	Alberta
666666668	Sunderland	James	7 Heartbroken Ave	Silent Hill	Alberta
620451097	Williams	Amanda	25 Rodeo Drive	Edmonton	Alberta
666666669	Wolf	Claudia	66 Twisted View	Silent Hill	Alberta
371988812	Carswell	Mary	425 Remington Ave	Calgary	Alberta

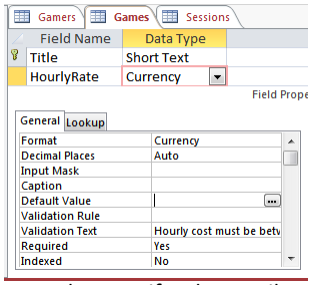
Primary Key for table 'Employees' is the 'SIN' attribute

Choosing A Primary Key

- A primary key must be unique to each record because it is the one thing that distinguishes them.
- If there's at least one instance where the attributes of two records can take on the same value then that attribute cannot be a primary key. (When in doubt verify with your users).
- If a primary key cannot be formed from a single attribute then several attributes can be combined into a composite key. (Each attribute is still a column but together they form a unique primary key for each record).
  - E.g., Course`Registrations` table: Course name, course number, lecture section (CPSC 203 L01)
- If a unique primary key still cannot be found then 'invent' one.
  - E.g., `StudentID#, SocialInsuraneNumber`

MS-Access: Views Of Your Database

- **Design view**



  - Used to specify what attributes that a table will consist of:
    - e.g., GAMES: `Title`, HourlyRate
  - Used to specify the type and the format and valid range of values
    - e.g., `SIN` is attribute with 9 characters that must be in the format 000 000 000
    - e.g., HourlyRate must be between $1 - $100

- **Datasheet view**

Title	HourlyRate
DOOMED	$7.00
EpicLegends	$10.00
FarmerTam	$6.00
FrankEsteinsHorror	$15.00
GrecoAncients	$20.00
LegendsOfLegend	$5.00
MindBlowingLegends	$20.00
Piraters	$13.00
TheTams	$20.00
WOWEE	$10.00

- Once the attributes have been specified in the Design view using the Datasheet view allows data entry for each record.

Example Problem: Online Games

- **This example can be found online:**
  - http://pages.cpsc.ucalgary.ca/~tamj/2015/203F/database/LectureExample.accdb
- An online gaming server will allow several online different games to be played
- Gamers can logon to play a particular game
- A gamer playing a game will create a ‘session’ that tracks (among other things) the cost of the gaming session

Online Gamers: Information To Be Tracked

- Online identifier: "Call sign"
- Contact information: Email
- Contact information: Telephone number
- Income: A (yearly) numeric figure
- Real life identifier: First and last name
- Overall 'score' (sum of player's accomplishments among multiple games): Level

Online Games: Information To Be Tracked

- Name of the game: Title
- The cost of playing a game: Hourly rate

Gaming Sessions: Information To Be Tracked

- Each time that a player starts playing a game billing information must be generated (allows the bill to be attached to the correct player)
  - Who played the game (who gets the bill)
  - Which game was played (how much is the cost per time unit)
  - How long was the game played (in conjunction with the cost per time unit it determines the size of the bill)

Picking Tables

- A table stores related information
  - E.g.,
  - Client: Client name, purchases, phone, address, email
  - Product: Product name, price, description
- The three groups of information in this problem appear to map to three database tables
  - Gamers
  - Games
  - Sessions

Guidelines For Naming Tables

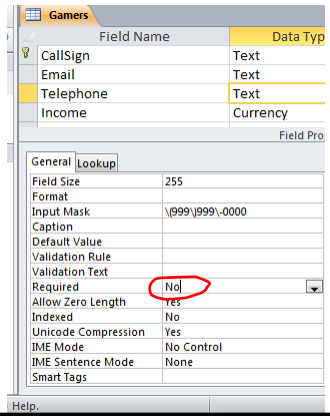
1. Create a unique and descriptive name.
   – "`VehicleMaintenanceRecords`" vs. "`CarDetails`"
2. Do not use words that convey database or technical terminology (use real world terms)
   – "`File`", "`Record`", "`Table`"
3. While names should be short avoid using acronyms and abbreviations unless they are well-known.
   – "SC" = ???
4. Consider using the plural form of a name.
   – "`Games table`" vs. "`Game table`"
5. Avoid the use of spaces in names.
   – "`Undergraduate students`" vs. "`Undergraduate_Students`" vs. "`UndergraduateStudents`"

Guidelines For Naming Attributes

1. Select a unique and descriptive name (similar to tables).
2. Create a name that accurately, clearly and unambiguously identifies the characteristic that the attribute represents.
   – "`Name`" vs. "FirstName"
3. While names should be short avoid using acronyms and abbreviations unless they are well-known (similar to tables).
4. Use the singular form of a name
   – Tables store multiple records (e.g., `GAMES` table), attributes store a single piece of information (e.g., `Title` for a particular game)
5. Avoid the use of spaces in names (similar to tables).

Null Values

- Refers to the attributes of a record that are empty
- Primary keys cannot be null but other attributes may be null
- Entry of any attribute can be made mandatory



Gamers Table: Attributes

- Gamer information to track:
  - Online identifier: "Call sign"
  - Contact information: Email
  - Contact information: Telephone number
  - Income: A (yearly) numeric figure
  - Real life identifier: First and last name
  - Overall score: Level

GAMERS

CallSign	Email	Telephone	Income	LastName	FirstName	Level

Games Table: Attributes

- Game information to track:
  - Name of the game: Title
  - The cost of playing a game: Hourly rate

GAMES

Title	HourlyRate

Sessions Table: Attributes

- Each time a player starts playing a game billing information must be generated.
  - Who played the game
  - Which game was played
  - How long was the game played
- This one is trickier!
  - Identifying 'who': need to be 100% certain that the correct gamer has been identified (don't bill the wrong person)
  - Identifying 'which': again certainty is required because different games have different hourly rates (don't bill for the wrong game and/or generate a bill for an incorrect amount)
  - We need to "hold off" on creating a table until the above two requirements can be met

Refinements Needed: Gamers

- Primary key?

GAMERS

CallSign	Email	Telephone	Income	LastName	FirstName	Level

Modified Table: Gamers

- Primary key: *CallSign*

GAMERS

CallSign	Email	Telephone	Income	LastName	FirstName	Level

Refinements Needed: Games

- Primary key?

GAMES

Title	HourlyRate

Modified Table: Games

- Primary key: Title

GAMES

Title	HourlyRate

The `Sessions` Table Revisited

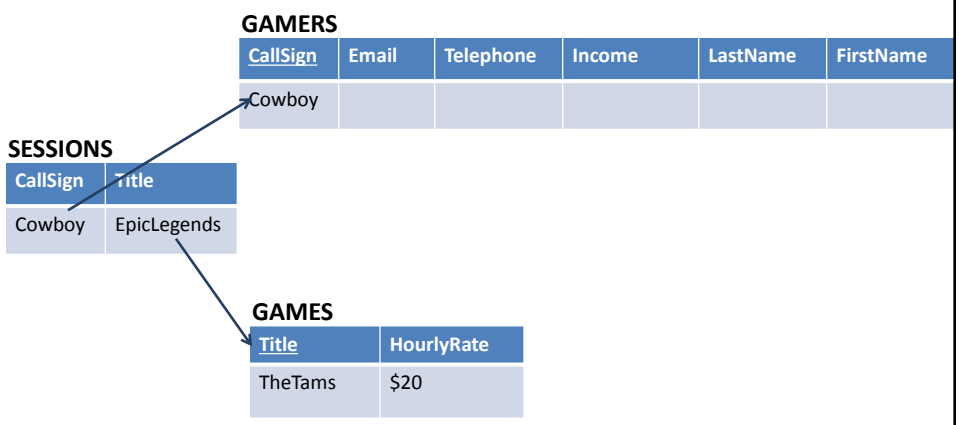
- Recall: Each time that a player logs in to play a game billing information must be generated.
- Some info need to generate a bill
  - Who played the game
  - Which game was played
- The 'who' needed to identify the gamer and the 'which' needed to specify the game
- Now that primary keys have been chosen for those two tables we can specify those two attributes

SESSIONS

CallSign	Title

Foreign Key

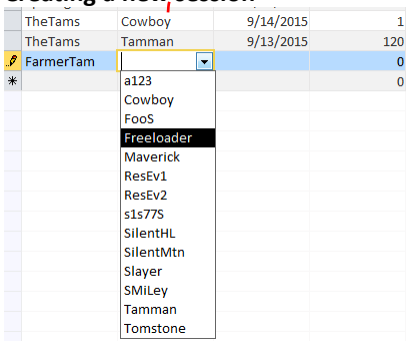
- An attribute in one table that refers to an attribute in another table:
  - E.g. `CallSign` in the `Sessions` table actually refers to a player's call sign in the `Gamers` table



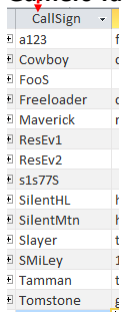
Purpose Of Foreign Keys

- Using foreign keys can prevent errors
- Example when we create a login playing session, we can ensure that we only bill a player that already exists in the `Gamers` table.

Creating a new session



Gamers Table



- (The same principle applies to the '`Title`' foreign key)

Refinements Needed: `Sessions`

- It's determined that each player can only login once per day.
- Players can login and play over multiple dates
- For each session we could store the login date and the duration (minutes):

SESSIONS

CallSign	Title	SessionDate	SessionDuration
Cowboy	TheTams	9/13/2015	120

Refinements Needed: Sessions

- Each row in the table is created when a gamer logins on a particular date
- Primary key?

SESSIONS

CallSign	Title	SessionDate	SessionDuration
Cowboy	TheTams	9/13/2015	120

Composite Key

- Reminder: It's a primary key that consists of multiple attributes (multiple columns in a database table)

Attribute1	Attribute2	Attribute3	Attribute4

Modified Table: `Sessions`

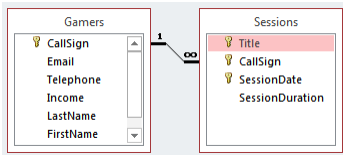
- Primary key (composite): `CallSign`, `Title`, `Date`
- The creation of the primary key 'makes sense' intuitively for this example based on the previous restrictions.

SESSIONS

CallSign	Title	SessionDate	SessionDuration
Cowboy	TheTams	9/13/2015	120

Relationships Between Tables

- Relationships occur when an attribute of one table is a foreign key in another table.



- Multiplicity: indicates how many instances of a particular item participates in the relationship:
  1. One to one
  2. One to many
  3. Many to many

Multiplicity

1. One to one relationships
   - One entity participates in the relationship from the 'left' and one entity participates in the relationship from the 'right'.
   - Person : Head
   - Gamers : CallSign
   - This type of relationship is rare in databases
2. One to many relationships
   - On one side of the relationship one entity participates in the relationship while on the other side: zero or more entities may participate in the relationship.
   - Person : Hair
   - Gamers : Sessions : Games

Multiplicity (2)

3. Many to many relationships
   - On each side of the relationship zero or more entities may participate in the relationship.
   - E.g., Travelers : Destinations

Travelers table

TravelerID	LastName	FirstName
1	Tam	James
2	Jones	Mary
3	Smith	Jon

Destinations table

DestinationID	DestinationName
1	Dubai
2	Paris
3	Cairo
4	Vulcan

Multiplicity (3)

- Many to many relationships
  - Typically implemented as two one to many relationships in databases:

Travelers table

TravelerID	LastName	FirstName
1	Tam	James
2	Jones	Mary
3	Smith	Jon

Destinations table

DestinationID	DestinationName
1	Dubai
2	Paris
3	Cairo
4	Vulcan

Trips table

TravelerID	DestinationID	Date
1	1	Sept 1 2015
2	3	Sept 1 2015
2	4	Sept 8 2015

Many To Many: Ignoring The Rule

Travelers table

TravelerID	LastName	FirstName
1	Tam	James
2	Jones	Mary
3	Smith	Jon

$Dest^1$	$Dest^2$	$Dest^3$	...	$Dest^n$
Dubai				
Dubai	Cairo	Vulcan		Zimbobway
NY	Vulcan			

Many To Many: Ignoring The Rule (2)

Destinations table

DestinationID	DestinationName
1	Dubai
2	Paris
3	Cairo
4	Vulcan

$Trav^1$	$Trav^2$	$Trav^3$	...	$Trav^n$
Alice	Bob	Bill		Zeek
Alice	Bill	Charlie		
Alice	Bill			
Jim	Karen			

(`Gamers` : `Games`) could be implemented as a many to many relationship (by-passing the Sessions table) but problems similar to the previous example would be encountered.

Primary-Foreign Keys Again

- When there is a one to many relationship the primary key of the 'one' side becomes a foreign key on the 'many' side.
- Examples:

1 — `Gamers` : `Sessions` — Many

CallSign: Primary key — **CallSign: Foreign key**

1 — `Games` : `Sessions` — Many

Title: Primary key — **Title: Foreign key**

Diagrammatically Representing Database Tables

- Entity-Relation diagrams (E-R Diagrams or E.R.D.'s): show the attributes of a table

Format

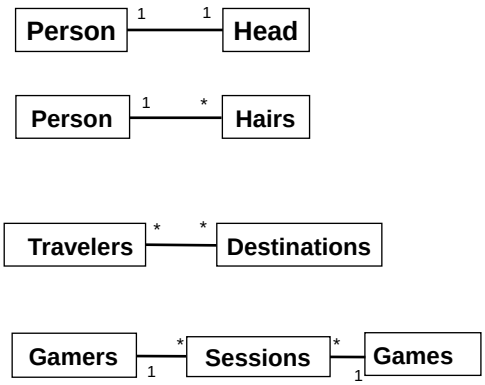
TABLE NAME
Primary key
Attribute
Attribute

Example

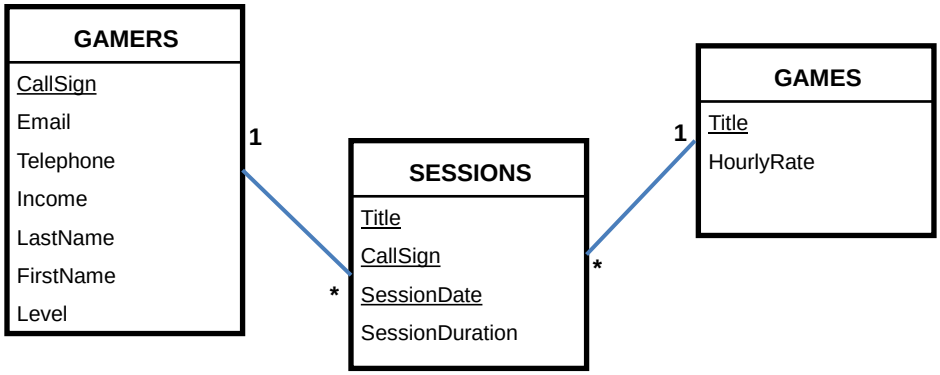
GAMES
Title
HourlyRate

Diagrammatically Representing Relationships

- ERDs Graphically represent relationships between tables as well as any enforced rules on multiplicity:



The ERD For The Example Database



Types Of Tables

- **Data tables**
  - Dynamic, will likely be manipulated over the life the database (add, delete, modify)
  - All three tables used in the example: `Gamers`, `Games`, `Session` are data tables
- **Validation tables**
  - Used to ensure data integrity (to 'lookup' values)
  - Typically it maps one value to another (e.g., product code to a product, an ISBN number to a book)
  - Rarely (if ever) changes
  - E.g., `Faculties` table (some of the codes actually used at the University of Calgary)

Code	DepartmentName
AR	Arts
HA	Haskayne School of Business
RO	Environmental Design
SC	Science

Example Use Of Data Table

- AR?
- ED?
- GS?
- HA?
- KN?
- MD?
- RO?
- SC?

- `Faculties` table:

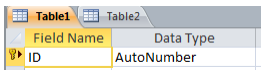
Code	FacultyName	Department1	Department2	…	Department 10
SC	Science	Biology	Chemistry	…	Mathematics & statistics
ED	Environmental Design	Mast. Architecture	Mast. Environmental Design	…	Mast. Landscape Architecture

- Other tables in our database would 'lookup' the code from the rows of this database
  - `Lectures` table:

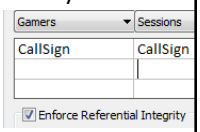
ID	LastName	FirstName	Faculty	Program	…
11 111 111	Tam	James	SC	CPSC	
11 111 112	Jones	Smith	ED	MEDES	

Types Of Data Integrity In Databases

1. Table-level integrity (entity integrity):
   - Ensuring that no duplicate records exist.
   - Implementation: no primary keys are null: MS-Access (automatic) indexed – no duplicates.



2. Relationship-level integrity (referential integrity):
   - Ensuring that relationship between a pair of tables is sound and the records in the tables are synchronized when data is entered into, updated in or deleted from either table (MS-Access: only partially implemented).
   - Implementation: use lookup values between tables



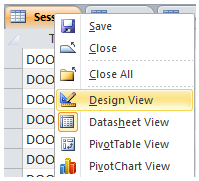
3. Field/attribute -level integrity (domain integrity):
   - Ensuring that the attributes are valid and accurate.
   - MS-Access implementation: input masks and validation rules.

Input Masks

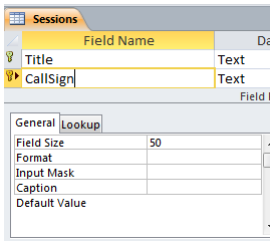
- Ensures the proper format for the data entered into the database
- Example: SIN number must be entered as:
  - *<three digits>* <space> *<three digits>* <space> *<three digits>*
  - Invalid inputs:
    - Abc def ghi
    - 321 22 4234
- Online example: Telephone number format
  - (*<area code>*)*<3 digits>*-*<4 digits>*
  - Example:
    - (403)210-9455

Defining Input Masks

- Switch to 'design view'

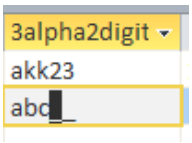


- Specify the required format under the 'Input mask' property of the appropriate table attribute

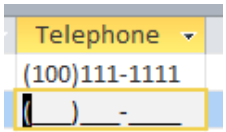


Use Of Input Masks

- Constrains input allowed
  - Can only enter a single digit
  - Can only enter a single character
  - Can only enter 5 digits (zip code)
  - Etc.
- "Ignores" invalid inputs in real-time



- Specifies the format of data to be entered (data entry cues)

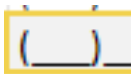


Some Characters That Define Input Masks

- Source (last accessed Sept 2015):
  - https://support.office.com/

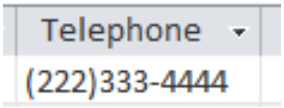
Desired input for each character	Character to enter as input mask
Digits only (0…9)	0 or 9
Digits, space (default), plus or minus sign	#
Alphabetic letter	L
Alphabetic letter or digit	A, a

  - To display a character literally in the input mask
    \<char>

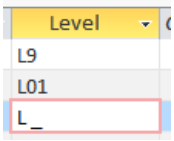


Input Masks: Online Database Example

- Gamers table: Telephone number

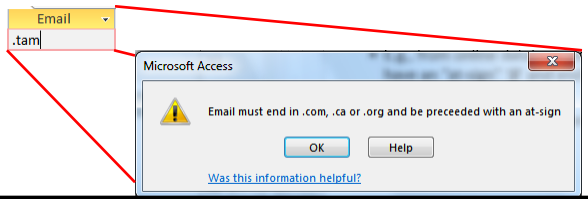


- Gamers table: Level



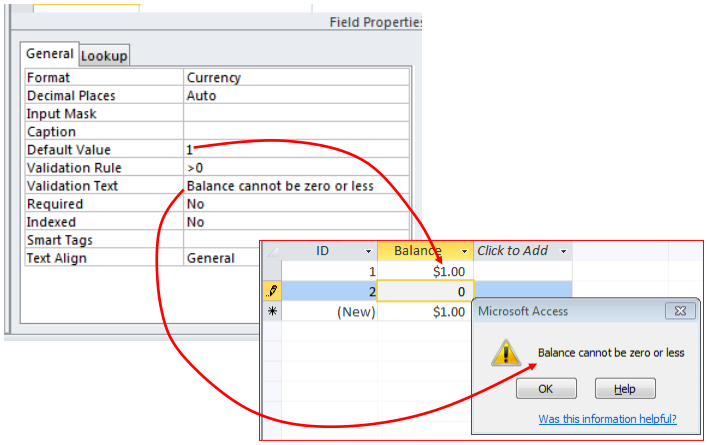
Validation Rules

- Validation rules check the data is in the valid range.
  - E.g., from online database example: `Gamers` table, `income` must be a non-negative value
- Can also be used to specify a data format
  - E.g., from online database example: `Gamers` table, a valid `email` must have an "at-sign" '@' and end in one of the following suffixes '`.ca`', '`.com`', "`.org`"
  - Unlike input masks validation rules allows useful error messages to be displayed



Validation Rules: Specifying Error Messages

- "Validation text" & "default values"



Example Database: Application Of The Validation Rules

- `Gamers` table
  - `CallSign`: first character must be alphabetic
  - `Email`: must include an 'at-sign'/@ and then end in '`.ca`', '`.com`' or '`.org`'
  - `Income`: no negative values
- `Games` table
  - `HourlyRate`: a dollar value from $1 to $100.
- `Sessions` table
  - `SessionDate`: date must be from Sept 12 2015 onwards
  - `SessionDuration`: specifies the number of seconds from 0 – 86,400

The **Wildcard**

- A value that can be used in place of other values.
- Example: “The joker is wild” option in card games
- Example: “`*.docx`” documents only can be used to specify that only documents ending in the suffix “`.docx`” with any name will be considered.
- The start character ‘*’ is a wildcard because it can be substituted by zero or more characters
  - Example documents that will considered
    - `resume.docx`
    - `A.docx`
  - Example documents that won’t be considered
    - `resume.doc`
    - `Me.jpg`
- The wildcard can be used in conjunction with validation rules

Validation Rules: Online Database Example

- `Gamers` table: `Income` (non-negative only)

Income
$12,000,000.00
$0.00

- `Gamers` table: `Level` (<L> <*one or two digits*>)

Level
L07
L12
L02
L

Validation Rules: Online Database Example (2)

- `Gamers` table: `CallSign` (first character must be alphabetic)
  - O.K.

    Tamman
    Tomstone
    zzephyr

  - `Not O.K.`

    1foo

- `Gamers` table: `Email` (at-sign followed by '`.com`', '`.ca`', '`.org`'
  - O.K.

    heather@morris.com
    harry@mason.com
    tam_yeah_right@hotmail.com

  - `Not O.K.`

    foo.com

Validation Rules: Online Database Example (3)

- `Games` table: `HourlyRate` ($1 - $100)
- `Sessions` table: `Duration` (Minutes in a day spent playing 0 – 86,400)

Documenting A Database

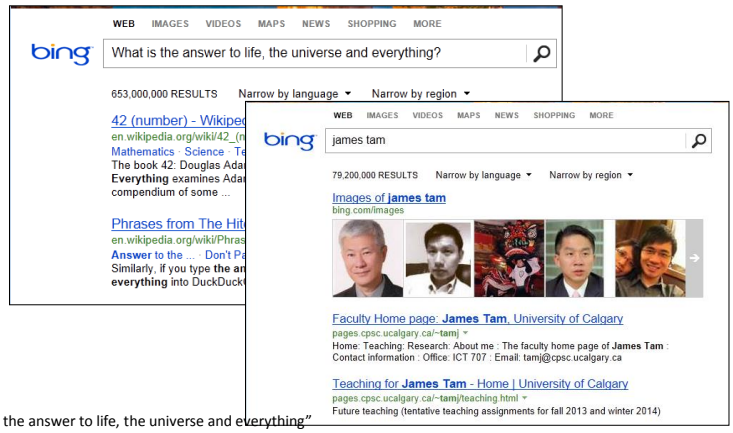
- Documentation: Provides information about the database to the other people who will be working on database.
- In MS-Access documentation can be entered in the "Description" column (under the Design view)

Field Name	Data Type	Description
CallSign	Text	Must begin with an alphabetic character, the rest can be anything
Email	Text	<anything> @ <anything> end in .com, .ca or .org
Telephone	Text	Format: (ddd)ddd-dddd
Income	Currency	Must not be a negative value
LastName	Text	
FirstName	Text	
Level	Text	<L> <One or two digits> e.g., L1, L99

- It can provide information about the type and format of the information to be stored.
  - Can be used if errors are found. (Providing the original 'intention' if there is an error in the validation rules or the input mask can help others correct the error).

Database Queries

- Queries are questions 'asked' of/to the database in order to retrieve information.

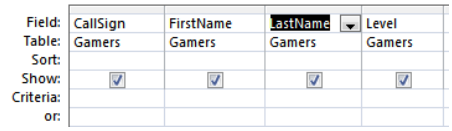


"What is the answer to life, the universe and everything"
– *The Hitchhiker's Guide to the Galaxy* (Del Rey 1979)
– *The Hitchhiker's Guide to the Galaxy* (BBC 1981)

Retrieving Data Via Queries

- Data retrieval occurs through the use of 'queries':
  - A query is a question asked of the data in the database.
  - May be worded to show only the parts of the database for which the answer to the question is true OR it be worded to just show the values of the attributes specified in the query
  - Example: What is the `CallSign`, `FirstName`, `LastName` and `Level` rate of every gamer in the `Gamers` Table:

Query (graphical form is an Access specific query)

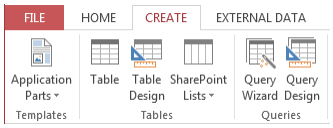


Result of the query

CallSign	FirstName	LastName	Level
a1			
a123	Mary	Carswell	L9
Aamazing			L01
Az			
Cowboy	Tough	Texan	L99
FooS	Maureen	Edgar	L1
Freeloader	...kidding me!	You gotta be..	L13
Maverick	John	Maverick	L77

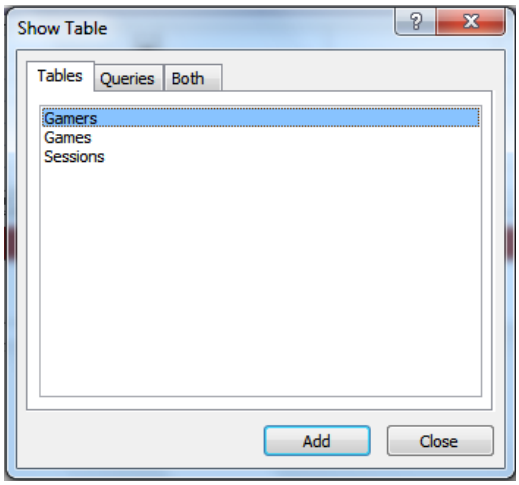
Forming Queries Graphically In Access

- Create->Query Design



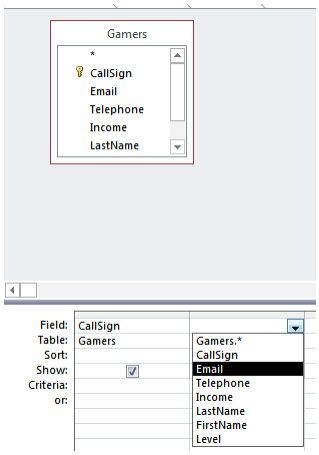
Forming Queries Graphically In Access (2)

- Select the desire table or tables



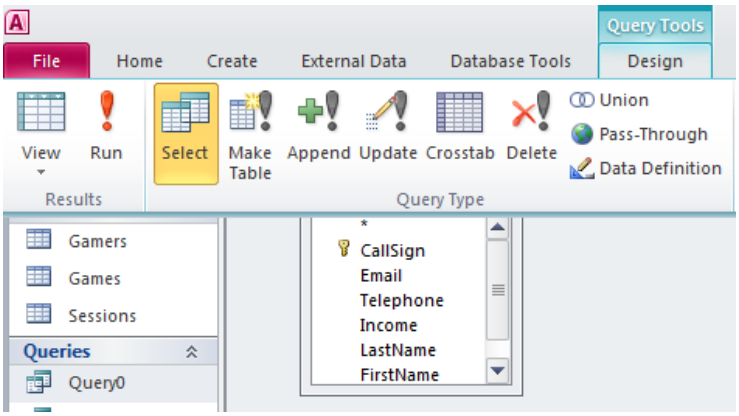
Forming Queries Graphically In Access (3)

- Select the attributes of the table



Forming Queries Graphically In Access (4)

- Run the query to view the results



Query #1: Specifying Query Criteria

- What is the `CallSign` & `Email` & and `FirstName` of all gamers that have the last name of Tam?

CallSign	Email	Telephone	Income	LastName	FirstName	Level
a1			$0.00			
a123	foo@bar.ca		$12,000,000.00	Carswell	Mary	L9
Aamazing			$0.00			L01
Az	a@b.com		$0.00			
Cowboy	countryboi@hot	(111)111-1111	$123,000.00	Texan	Tough	L99
FooS			$42,500.00	Edgar	Maureen	L1
Freeloader	cheap@skate.or		$0.00	You gotta be..	...kidding me!	L13
Maverick	rebel@yell.ca	(222)333-4444	$75,000.00	Maverick	John	L77
ResEv1			$35,000.00	Keddney	Leon	L14
ResEv2			$42,000.00	Redfeld	Claire	L15
s1s77S			$0.00	Jones	Mary	L25
SilentHL	heather@morris	(403)210-9455	$6,500.00	Maurice	Heather	L17
SilentMtn	harry@mason.cc	(403)210-9455	$55,000.00	Masoon	Harri	L43
Slayer	tam_yeah_right(	(123)456-7890	$100,000.00	Tam	James	L88
SMiLey	1@1.com	(222)222-3333	$1.00	Wang	Tam	L07
Tamman	tama@aol.com		$55,000.00	Tam	Tam	L12
Tomstone	gm ail@gmail.co	(403)111-2222	$75,000.00	Torrie	Donald	L02
zzephyr	1@*.com	(100)111-1111	$0.00			

CallSign	Email	FirstName
Slayer	tam_yeah_right@hotmail.com	James
Tamman	tama@aol.com	Tam

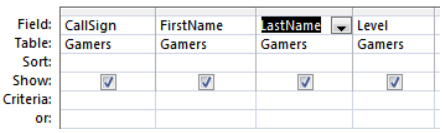
SQL (Structured Query Language)

- It's the universal language for querying a database (very widely used!)
- Unlike graphical queries the statements are portable between different database programs.
- Queries are formed using text descriptions (can be more powerful but more complex than graphical queries):
  - **SELECT**: Specifies the relevant attributes of which tables are involved in the query e.g., `CallSign` attribute of the `Gamers` table
  - **FROM**: Lists the tables from which the data is to be selected
  - **WHERE**: Provides the conditions to determine if a particular row shows or doesn't show as a result
  - **ORDER BY**: Specifies the order in which rows are to be returned**;**

Note: Capitalizing of the above four words is a standard SQL convention.

Example SQL Query

- Example: What is the `CallSign`, `FirstName`, `LastName` and `Level` of every gamer in the `Gamers` Table:
- **Graphical Access query:**



- **SQL:**

```
SELECT Gamers.CallSign, Gamers.FirstName,
       Gamers.LastName, Gamers.Level
FROM Gamers;
```

CPSC 203: Forming Queries

- You need to know how to form and evaluate queries graphically or using SQL

Queries Can Span Multiple Tables

- This is referred to as a 'join' because the results join data from multiple tables

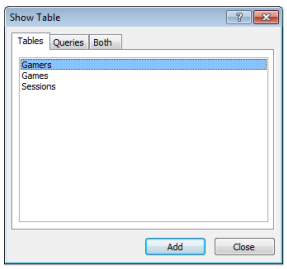
Query #2: Multi-Table Queries

- What is the: `CallSign` (`Gamers`), `Title` (Games) and `Date` (`Sessions`) of every game played by a gamer our database.
  - A gamer must have played at least one game (created a game session)
  - A game must have been played by at least one gamer

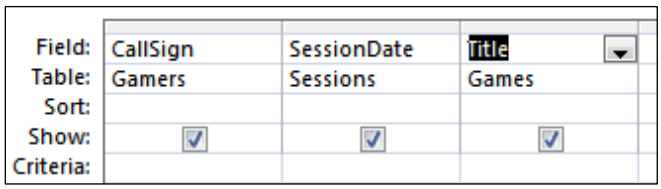
CallSign	SessionDate	Title
SMiLey	9/13/2015	DOOMED
Tamman	9/13/2015	TheTams
a123	9/13/2015	EpicLegends
Cowboy	9/13/2015	DOOMED
Tomstone	9/16/2015	DOOMED
Tomstone	9/14/2015	DOOMED
Tomstone	9/17/2015	DOOMED
Cowboy	9/14/2015	TheTams
Freeloader	9/13/2015	DOOMED
Freeloader	9/14/2015	DOOMED
Freeloader	9/15/2015	DOOMED
Freeloader	9/16/2015	DOOMED
Freeloader	10/31/2015	DOOMED
Freeloader	11/17/2015	DOOMED

Query #2: Access

- Add the desired tables to the query



- Select the attributes of these tables relevant to the query



Query #2: SQL

- Auto generated form from Access
  - `SELECT Gamers.CallSign, Sessions.SessionDate, Games.Title`
  - `FROM Games INNER JOIN (Gamers INNER JOIN Sessions ON Gamers.CallSign = Sessions.CallSign) ON Games.Title = Sessions.Title;`
- Acceptable form for this class
  - `SELECT Gamers.CallSign, Sessions.SessionDate, Games.Title`
  - `FROM ((Games JOIN Gamers JOIN Sessions) ON (Gamers.CallSign = Sessions.CallSign ON Games.Title = Sessions.Title));`

Calculated Values

- In Access they are attributes /columns of the query that are derived from one or more attributes/columns of the tables used in the query

GAMES Table

Title	HourlyRate
TheTams	$20

(In the query this value is derived from the hourly rate)
RatePerMinute:
[HourlyRate] / 60

Query #3: Calculated Values

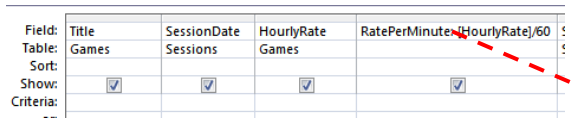
- Show the `Title`, `SessionDate`, `HourlyRate`, `RatePerMinute`, `SessionDuration` and the `SessionCost` of games that were played.
- Note: A conversion is needed
  - Cost for playing a game is cost/hour

Title	HourlyRate
DOOMED	$7.00

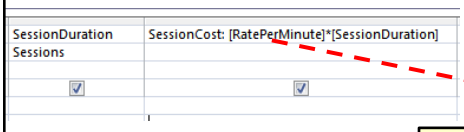
  - Session duration is listed in minutes so the cost is converted from cost/hour to cost/minute (divide hourly rates in '`Games`' by 60).

Query # 3: Graphical Access Query

- "**Calculated Values**" were used: The results of some columns were not stored in tables but derived from the values in other columns

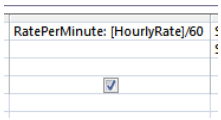


RatePerMinute:
`[HourlyRate] / 60`

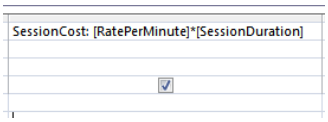


SessionCost:
`[RatePerMinute] * [SessionDuration]`

Query #3: SQL Form



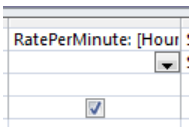
- `SELECT Games.Title, Sessions.SessionDate, Games.HourlyRate, [HourlyRate]/60 AS RatePerMinute, Sessions.SessionDuration, [RatePerMinute]*[SessionDuration] AS SessionCost`



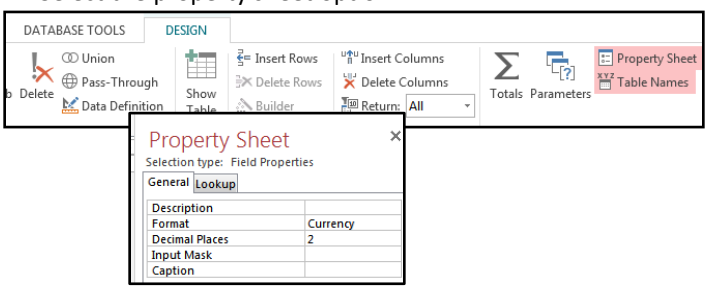
- `FROM Games JOIN Sessions ON Games.Title = Sessions.Title;`

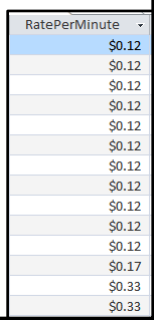
Access: Cleaning Up The Results

- Select the attribute



- Ribbon: Select the design tab,
  - Select the property sheet option



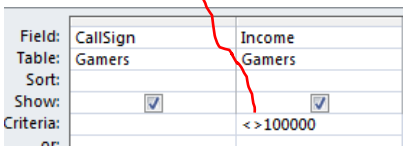


Logical Operators

Operation	Description	MS-Access operator
AND	•All conditions must be true for the result to be true. •If any condition is false then the entire result is false.	And
OR	•All conditions must be false for the result to be false. •If any condition is true then the entire result is true.	Or

Relational Operators

Operator	Description
<	Less than
<=	Less than , equal to
>	Greater than
>=	Greater than, equal to
=	Equal to
<>	Not equal to

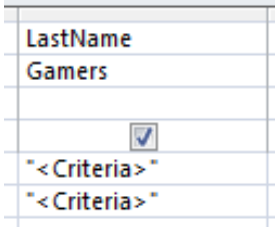


Query #4: Logical-OR

- Show the `CallSign`, `LastName` & `FirstName` of all employees whose last name is “Maurice” or “Masoon”.

CallSign	LastName	FirstName
SilentMtn	Masoon	Harri
SilentHL	Maurice	Heather

- Note that query criteria specified *within a column* will have the logical OR operation applied.

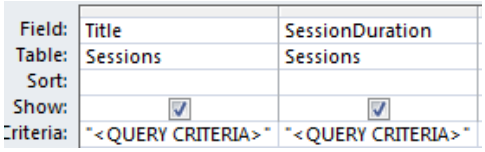


Query #5: Logical-AND

- Show all online sessions of the game title “DOOMED” which had a session lasting an hour or more.

Title	SessionDuration
DOOMED	60
DOOMED	1200
DOOMED	300
DOOMED	360
DOOMED	12000

- Note that query criteria specified *between columns* will have the logical AND operation applied to them.



Ordering Queries

- Query results can be ranked according to the attributes of a table.

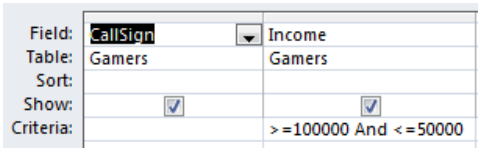
LastName	FirstName	CallSign
		a1
		Aamazing
		Az
		zzephyr
Carswell	Mary	a123
Edgar	Maureen	FooS
Jones	Mary	s1s77S
Keddney	Leon	ResEv1
Masoon	Harri	SilentMtn
Maurice	Heather	SilentHL
Maverick	John	Maverick
Redfeld	Claire	ResEv2
Tam	James	Slayer
Tam	Tam	Tamman
Texan	Tough	Cowboy
Torrie	Donald	Tomstone
Wang	Tam	SMiLey
You gotta be..	...kidding me!	Freeloader

Query #6: Ordering Queries

- Show the `CallSign`, `LastName` and `FirstName` of all the gamers in ascending order sorted first by last name, then by first name and finally by call sign.
- Use the "SORT" property (MS-Access) "`ORDER BY`" property (SQL)

Query #7: Contradictory Conditions

- Take care not to specify queries that can never be true!
- Example:
  - Show the `CallSign` and `Income` of all gamers with an income $50,000 or lower AND $100,000 or higher.




```
SELECT Gamers.CallSign,
Gamers.Income
FROM Gamers
WHERE (Gamers.Income >=100000 And
                Gamers.Income
       <=50000);
```

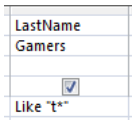
Using The **Wildcard** In Queries

- Similar to validation rules, the wild card can be used in queries when only partial information is known.
- Example: last name schwar*
- Use the '`Like`' operator under '`Criteria`' (MS-Access) or '`Where`' (SQL)

Wild Card: Example Queries

- Show last name when `LastName` begins with '`t`'

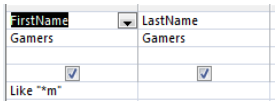
LastName
Texan
Torrie
Tam
Tam




```
SELECT Gamers.LastName
FROM Gamers
WHERE
  (Gamers.LastName Like "t*");
```

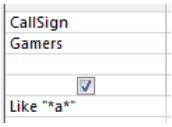
- Show first and last name when `FirstName` ends with '`m`'

Tam	Wang
Tam	Tam



- Show the call sign when there's an 'a' anywhere (call sign).

CallSign
Freeloader
a123
Aamazing
Az
a1
Maverick
Slayer
Tamman



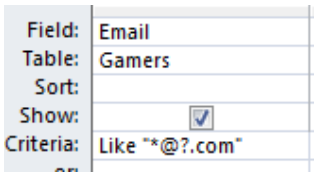
Single Character Wildcard '?'

- Show .com emails with only a single character between the '@' and the '.com'

Table to query:

Email
foo@bar.ca
a@b.com
countryboi@hotmail.com
cheap@skate.org
rebel@yell.ca
heather@morris.com
harry@mason.com
tam_yeah_right@hotmail.com
1@1.com
tama@aol.com
gm ail@gmail.com
1@*.com

Query design



After This Section You Should Now Know

- How a database is broken down into tables and how tables are broken down into it's component parts
- What are the type of tables and the purpose of each
- What is the purpose of a primary key
- What is a foreign key
- When table are related what is the rule for determining which table contains the primary vs. foreign key
- What is a null value
- What are forms of data integrity in databases
- Guidelines for naming tables and the attributes of the tables
- What are the three relationships that may exist between tables and how they differ

After This Section You Should Now Know (2)

- How is a many-to-many relationship typically implemented in a database
- The ERD representation of databases
- How to form different queries in order to retrieve data from a database
- How contradictions can result in no query results
- How wildcards can be used in queries