INFO20003 Database Systems

Dr Renata Borovica-Gajic*

Lecture 18
Database Administration

Week 9

slides adopted from David Eccles

Today's Lecture…

- Functions that are part of the DBA role
  - Capacity planning
    - Estimating disk space and transaction load
  - Backup and recovery
    - Types of failures, responses to these, types of backups

Capacity Planning

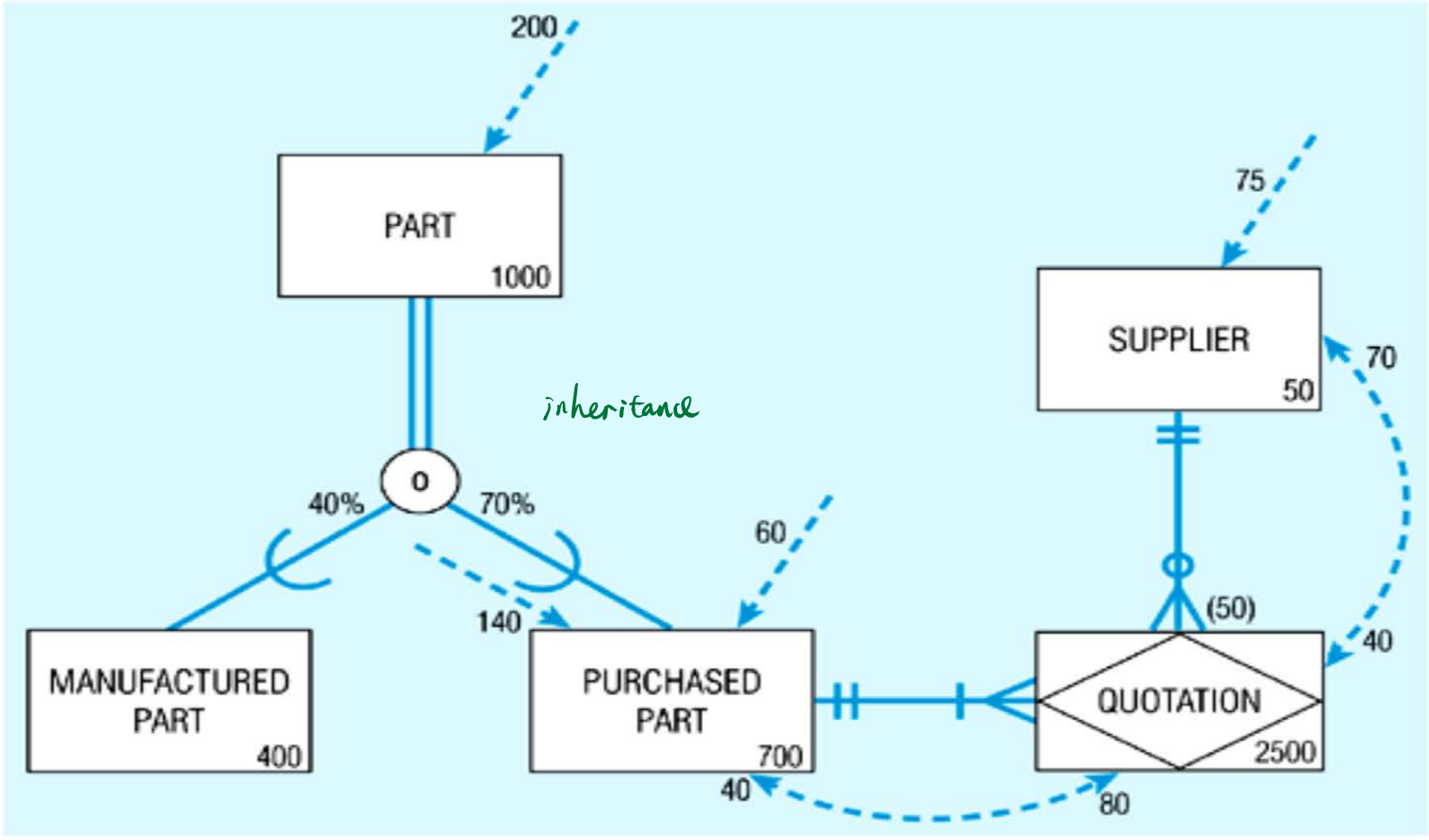
Capacity Planning

"Capacity Planning is the process of predicting when future load levels will saturate the system and determining the most cost-effective way of delaying system saturation as much as possible."

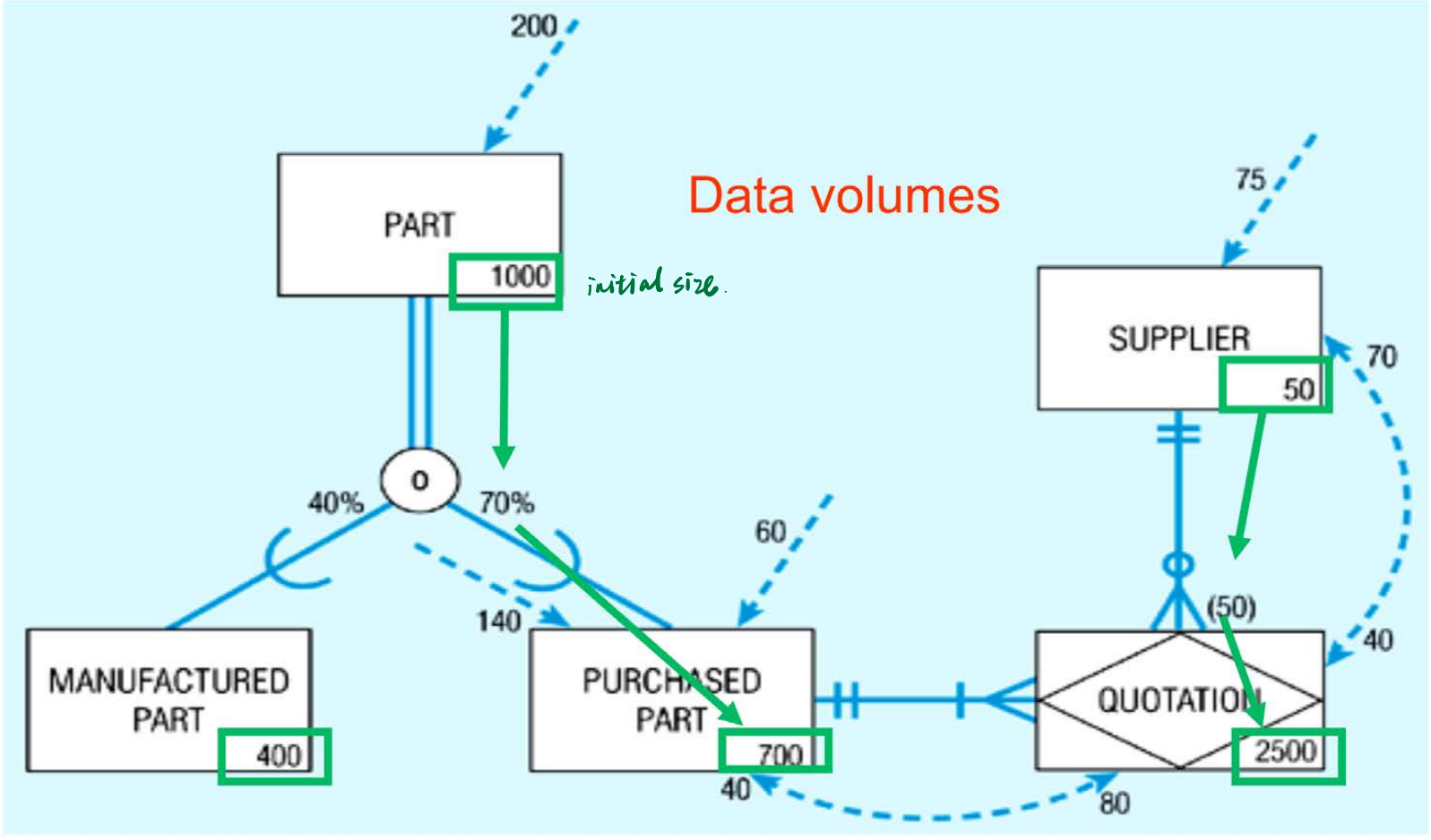
– Menasce and Virgilio (2002) 'Capacity Planning for Web Services'. Prentice Hall.

- When implementing a database, need to consider:
  - disk space requirements
  - transaction throughput → throughput : number of transaction per second / minute ... how often we are accessing the database
  - at go-live and throughout the life of the system
    - E.g. plan for 7 years can be 20 years

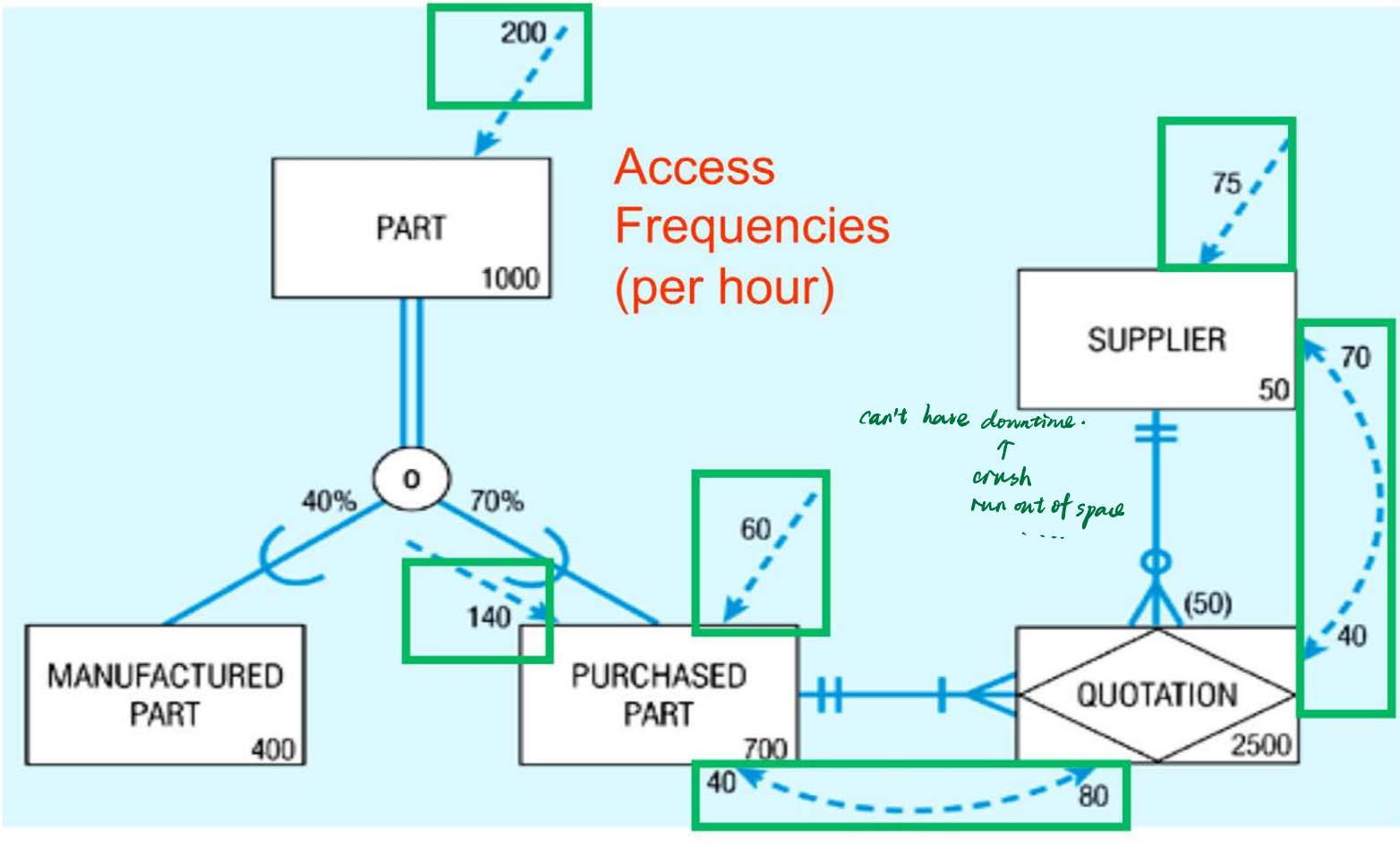
Estimating Database Usage



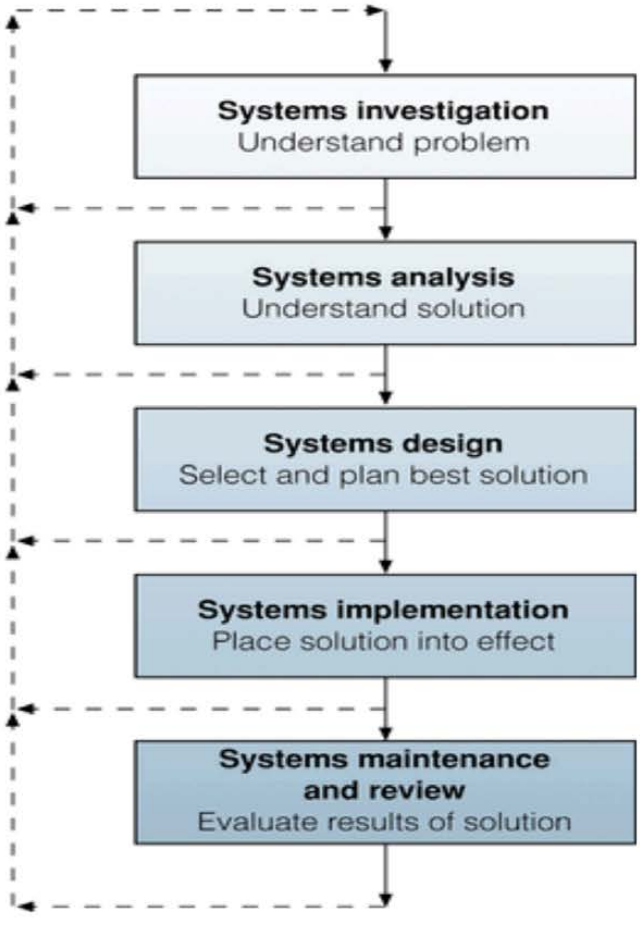
Estimating Database Usage



Estimating Database Usage



Capacity Planning in the dev life cycle



calculate expected disk usage and transaction load

monitor and predict disk usage and transaction load

Estimating disk space requirements

- Which estimation methodology to use?
  - many vendors sell capacity planning solutions
  - most have the same ideas at their core
  - here we present the core concepts
- Treat database size as the sum of all table sizes
  - Table size = number of rows * average row width

width (size of each record)

choose data type carefully ⇒ otherwise explode

Id	PostedBy	Forum	Content	ParentPost	WhenPosted
1	4	NULL	April is the cruellest month, breeding	4	2015-07-23 11:00:00
2	4	3	Lilacs out of the dead land, mixing	NULL	2015-03-11 11:00:00
3	3	NULL	Memory and desire, stirring	17	2014-11-04 11:00:00
4	3	NULL	Dull roots with spring rain.	68	2015-07-29 11:00:00
5	3	NULL	Winter kept us warm, covering	38	2014-11-30 11:00:00
6	3	NULL	Earth in forgetful snow, feeding	75	2015-06-29 10:00:00
7	3	NULL	A little life with dried tubers.	6	2015-06-07 10:00:00
8	5	NULL	Summer surprised us, coming over the Starnber...	76	2015-07-20 10:00:00
9	5	NULL	With a shower of rain; we stopped in the colonn...	21	2014-12-03 11:00:00
10	4	3	And went on in sunlight, into the Hofgarten,	NULL	2015-07-21 10:00:00

height

Calculating row widths

- Use information about storage sizes of different data types: https://dev.mysql.com/doc/refman/8.0/en/storage-requirements.html

Storage Requirements for Numeric Types

Data Type	Storage Required
TINYINT	1 byte
SMALLINT	2 bytes
MEDIUMINT	3 bytes
INT, INTEGER	4 bytes
BIGINT	8 bytes
FLOAT(*p*)	4 bytes if 0 <= *p* <= 24, 8 bytes if 25 <= *p* <= 53
FLOAT	4 bytes
DOUBLE [PRECISION], REAL	8 bytes
DECIMAL(*M*, *D*), NUMERIC(*M*, *D*)	Varies; see following discussion
BIT(*M*)	approximately (*M*+7)/8 bytes

eg. Age. → TINYINT

if use INT, waste a lot of memory

Calculating row widths

- These sizes are for MySQL and are slightly different for other vendors: https://dev.mysql.com/doc/refman/8.0/en/storage-requirements.html

Storage Requirements for Date and Time Types

Data Type	Storage Required
DATE	3 bytes
TIME	3 bytes
DATETIME	8 bytes
TIMESTAMP	4 bytes
YEAR	1 byte

→ precise

Calculating row widths

- https://dev.mysql.com/doc/refman/8.0/en/storage-requirements.html
  For VARCHAR/BLOB we use the average size (from catalog)

generally 1 char – 1 byte

CHAR(100) → 100 bytes

VARCHAR(100)

need to know average size

Storage Requirements for String Types

In the following table, *M* represents the declared column length in characters for nonbinary string types and bytes for binary string types. *L* represents the actual length in bytes of a given string value.

Data Type	Storage Required
`CHAR(M)`	*M* × *w* bytes, 0 <= *M* <= 255, where *w* is the number of bytes required for the maximum-length character in the character set. See Section 14.6.3.12.5, "Physical Row Structure" for information about `CHAR` data type storage requirements for `InnoDB` tables.
`BINARY(M)`	*M* bytes, 0 <= *M* <= 255
`VARCHAR(M)`, `VARBINARY(M)`	*L* + 1 bytes if column values require 0 − 255 bytes, *L* + 2 bytes if values may require more than 255 bytes
`TINYBLOB`, `TINYTEXT`	*L* + 1 bytes, where *L* < 2^8
`BLOB`, `TEXT`	*L* + 2 bytes, where *L* < 2^{16}
`MEDIUMBLOB`, `MEDIUMTEXT`	*L* + 3 bytes, where *L* < 2^{24}
`LONGBLOB`, `LONGTEXT`	*L* + 4 bytes, where *L* < 2^{32}
`ENUM('value1','value2',...)`	1 or 2 bytes, depending on the number of enumeration values (65,535 values maximum)

Estimate growth of tables

- How will tables grow over time?
- Gather estimates during system analysis, e.g.
  - "The company sells 1000 products. There are 2,000,000 customers who place, on average, 5 orders each per month. An average order is for 8 different products."

in database, table which increase in size frequently are called event table

therefore:
the Product table has 1000 rows.
the Customer table has 2,000,000 rows.
the Orders table grows by 10,000,000 rows per month.
the OrderItems table grows by 80,000,000 rows per month.

5 × 2,000,000

8 × 5 × 2,000,000

Estimate growth of tables

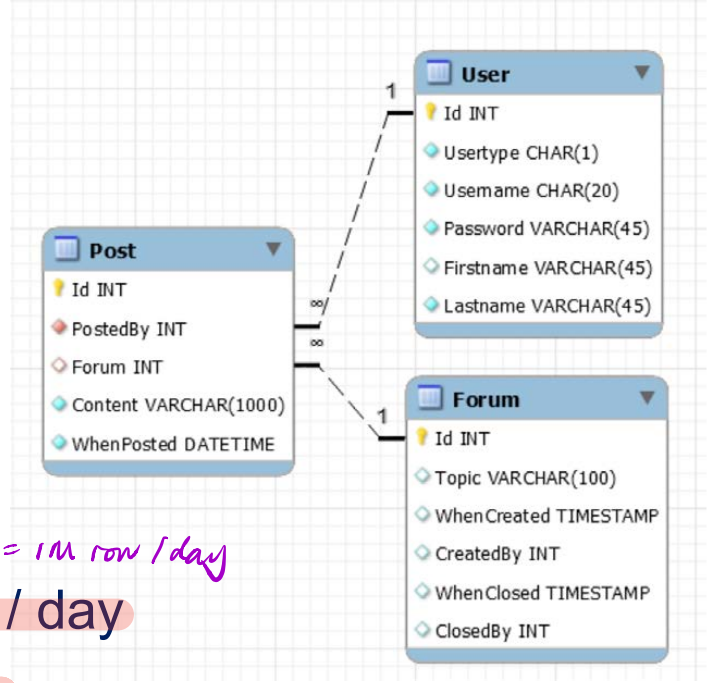
- Using this simplified database as an example, assume there are:
  - 100 forums
  - 1 million users

and assume that:

  - users post average 30 times per month

we calculate:

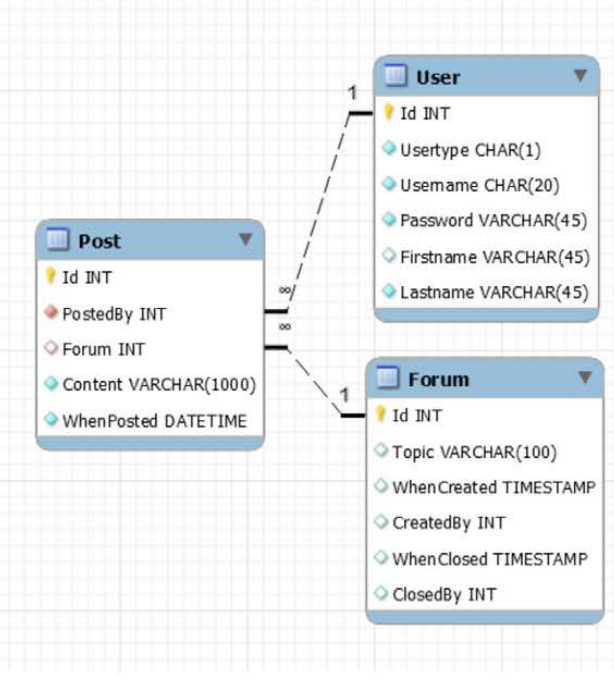
  - Post table grows by 1M rows / day
  - This is 12 inserts per second



$1M \times \frac{30}{30} = 1M$ row / day

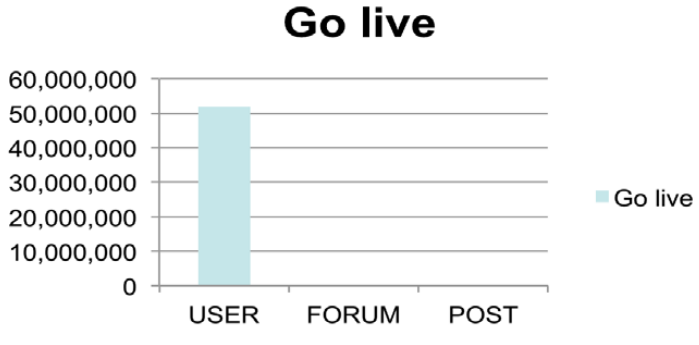
Calculate disk space per table

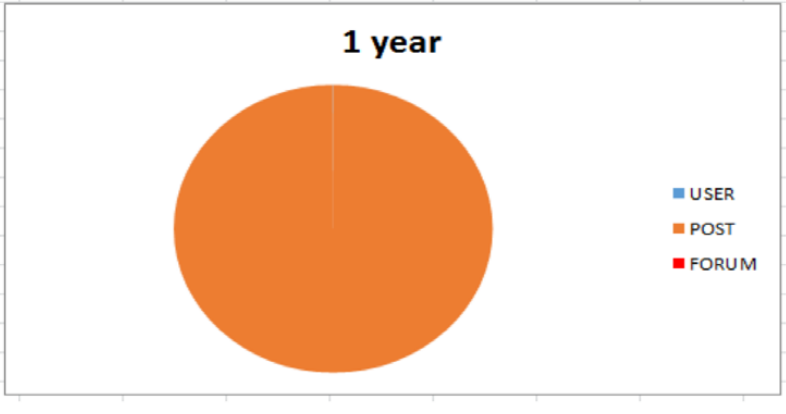
column	type	width	rows	1 month	1 year
USER					
Id	int	4			
UserType	char(1)	1			
UserName	char(10)	10			
Password	char(10)	10			
FirstName	varchar(45)	12	→avg		
LastName	varchar(45)	15	→avg		
ROW WIDTH		**52**	1,000,000	1,100,000	2,000,000
DISK SPACE			52,000,000	57,200,000	**104,000,000**
FORUM					
Id	int	4			
Topic	varchar(100)	50		per month	
WhenCreated	timestamp	4		1	
CreatedBy	int	4			
ClosedBy	int	4			
ROW WIDTH		**66**	100	101	113
DISK SPACE			6,600	6,666	**7,458**
POST					
Id	bigint	8			
PostedBy	int	4		per user per month	
Forum	int	4		30	
Content	varchar(1000)	500			
WhenPosted	datetime	8			
ROW WIDTH		**524**	0	30,000,000	390,000,000
DISK SPACE			0	15,720,000,000	**204,360,000,000**



Projected total storage requirements

Table	*Row width*	*No. rows at 1 year*	*Size*
User	52 bytes	2,000,000	104 Mb
Forum	66 bytes	113	0.007 Mb
Post	524 bytes	390,000,000	204 Gb
		TOTAL ->	*204 Gb*





Estimating transaction load

- Consider each business transaction
  - how often will transaction each be run?
  - for each transaction, what SQL statements are being run?

- For example, consider this fictitious banking application:

Transaction	Selects	Inserts	Updates	Delete	SQL/tr	Tr/cust/month	SQL/month	SQL/second
Withdraw	1	1	1		3	20	60,000,000	23
Deposit		1	1		2	5	10,000,000	4
Transfer	1	1	2		4	8	32,000,000	12
								39
no. customers		1,000,000						

DBs store so much more

- Under the hood there is so much more stored

when estimate,
also need to consider index

```
tab 0, row 1, @0x766
tl: 44 fb: --H-FL-- lb: 0x0 cc: 5 ==> Complete row (Head,First,Last)
col  0: [ 4]  52 4f 57 32        ==> VarChar2 data.
col  1: [13]  54 52 41 49 4c 49 4e 47 20 4e 55 4c 4c
col  2: [ 7]  77 c4 01 01 01 01 01 ==> DATE data.
col  3: [ 2]  c4 02              ==> Number data.
col  4: [10]  31 2c 30 30 30 2c 30 30 30 20
==>     ^^^^ Note: Size here is the column length for this PIECE, not the COLUMN.

==> Note there are only five columns - trailing columns are assumed NULL

tab 0, row 2, @0x320
tl: 1020 fb: --H-FL-- lb: 0x1 cc: 6 ==> 1020 bytes long, locked, 6 cols
col  0: [ 4]  52 4f 57 33
col  1: [11]  42 49 47 20 50 41 44 44 49 4e 47
col  2: [ 7]  77 c4 0c 1f 01 01 01
col  3: [ 2]  c1 02
col  4: [10]  31 20 20 20 20 20 20 20 20 20 ==> CHAR data. Note trailing blanks.
col  5: [975]
 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58
 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58
.... Repeated many times.
```

- Capacity planning is a rough estimation

Backup and Recovery

What is a Backup

- A backup is a copy of your data
  - however there are several types of backup
- If data becomes corrupted or deleted or held to ransom it can be restored from the backup copy
- A backup and recovery strategy is needed
  - To plan how data is backed up
  - To plan how it will be recovered

Protect data from …

- human error
  - e.g. accidental drop or delete
  - example: http://www.theaustralian.com.au/australian-it/human-error-triggered-nab-software-corruption/story-e6frgakx-1225962953523

- hardware or software malfunction
  - bug in application
  - hard drive (failure or corruption)
  - CPU
  - memory

Must also protect against

- malicious activity
  - security compromise
    - server, database, application
- natural or man made disasters
  - consider the *scale* of the damage
- government regulation
  - historical archiving rules
  - Metadata collection (AUS)
  - HIPPA, EU data retention regulations
  - Privacy Rules

Security

Texas cops lose evidence going back eight years in ransomware attack

We have to get very, very tough on cyber and cyber warfare… and backups?

By Alexander J Martin 27 Jan 2017 at 16:57 36 SHARE ▼

♫ I hacked the sheriff, but I did not hack his deputy ♫

Updated Cockrell Hill, Texas has a population of just over 4,000 souls and a police force that managed to lose eight years of evidence when a departmental server was compromised by ransomware.

In a public statement, the department said the malware had been introduced to the department's systems through email. Specifically, it arrived "from a cloned email address imitating a department issued email address" and after taking root, requested 4 Bitcoin in ransom, worth about $3,600 today, or "nearly $4,000" as the department put it.

Categories of Failure

Failures can be divided into the following categories:

- Statement failure → incorrect sql statement
  - Syntactically incorrect
- User Process failure (corruption, out of failure ...)
  - The process doing the work fails (errors, dies)
- Network failure → rebuild connection
  - Network failure between the user and the database
- User error → backup tool
  - User accidentally drops the rows, table, database
- Memory failure
  - Memory fails, becomes corrupt
- Media Failure
  - Disk failure, corruption, deletion

Types of Backups

- Physical vs Logical
- Online vs Offline
- Full vs Incremental
- Onsite vs Offsite

Physical backup

- Physical backup
  - raw copies of files and directories
  - suitable for large databases that need fast recovery
  - database is preferably offline ("cold" backup) when backup occurs
    - MySQL Enterprise automatically handles *file* locking, so database is not wholly off line
  - backup = exact copies of the database directories and files
  - backup should include logs
  - backup is only portable to machines with a similar configuration
  - to restore
    - shut down DBMS
    - copy backup over current structure on disk
    - restart DBMS

Logical backup

- Logical backup
  - backup completed through SQL queries
  - slower than physical
    - SQL Selects rather than OS copy
  - output is larger than physical
  - doesn't include log or config files
  - machine *independent*
  - server is *available* during the backup
  - in MySQL can use the backup using
    - Mysqldump
    - SELECT … INTO OUTFILE
  - to restore
    - Use mysqlimport, or LOAD DATA INFILE within the mysql client

DDL, DML statement

command will create database
CREATE table
INSERT table
index

Online vs Offline backup

- Online (or HOT) backup
  - backups occur when the database is "live"
  - clients don't realise a backup is in progress
  - need to have appropriate locking to ensure integrity of data

- Offline (or COLD) backup
  - backups occur when the database is stopped
  - to maximize availability to users take backup from replication server not live server
  - simpler to perform
  - cold backup is preferable, but not available in all situations e.g. applications without downtime → can't access for a time period

Drawback

Full vs Incremental backup

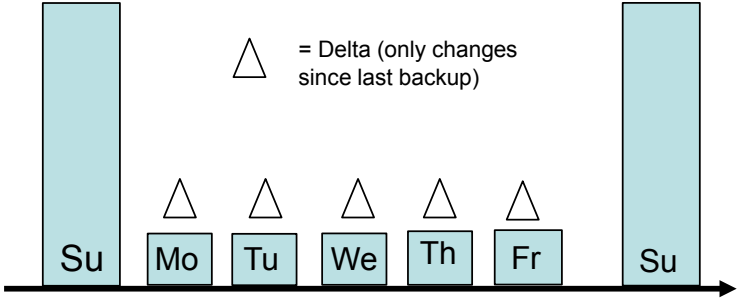
- Full
  - a full backup is where the complete database is backed up
    - may be Physical or Logical, Online or Offline
  - it includes everything you need to get the database operational in the event of a failure

Drawback: big, long time

- Incremental
  - only the changes since the last backup are backed up
  - for most databases this means only backup log files
  - to restore:
    - stop the database, copy backed up log files to disk
    - start the database and tell it to redo the log files

Create a Backup Policy

- Backup strategy is usually a combination of full and incremental backups
- For example:
  - weekly full backup, weekday incremental backup



- Conduct backups when database load is low
- If using replication, use the mirror database for backups to negate any performance concerns with the primary database
- TEST your backup before you NEED your backup (crucial)

Offsite Backup

- Enables *disaster recovery*
  (because backup is not physically near the disaster site)
- Example solutions:
  - backup tapes transported to underground vault
  - remote mirror database maintained via replication
  - backup to Cloud (see figure below)

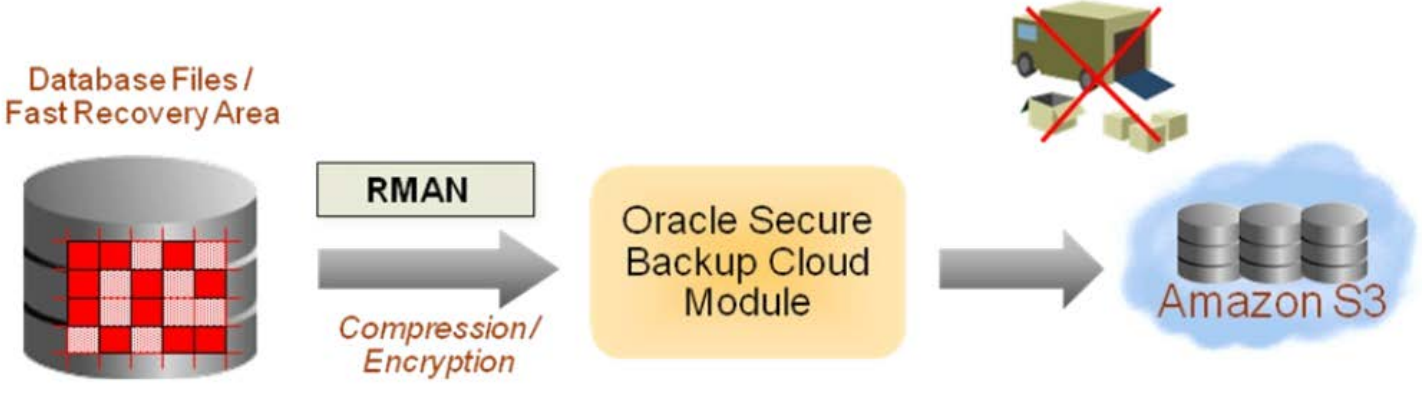


Figure 1. Oracle Database backup in the Cloud

What is examinable?

- The roles of a DBA
  - Capacity planning
    - Calculating Capacity & Transaction load
  - Back up and Recovery
    - Types of Failures
    - Backup Types

Next lectures

- Database warehousing