Dr Greg Wadley

THE UNIVERSITY OF MELBOURNE

INFO90002 Database Systems & Information Modelling

Week 09

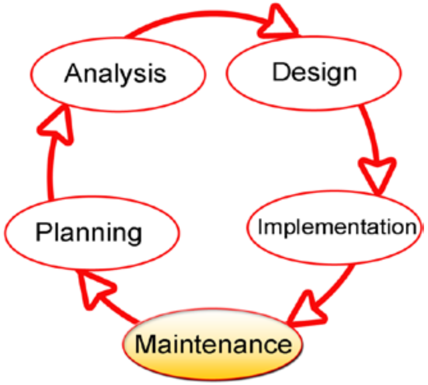
Database Administration

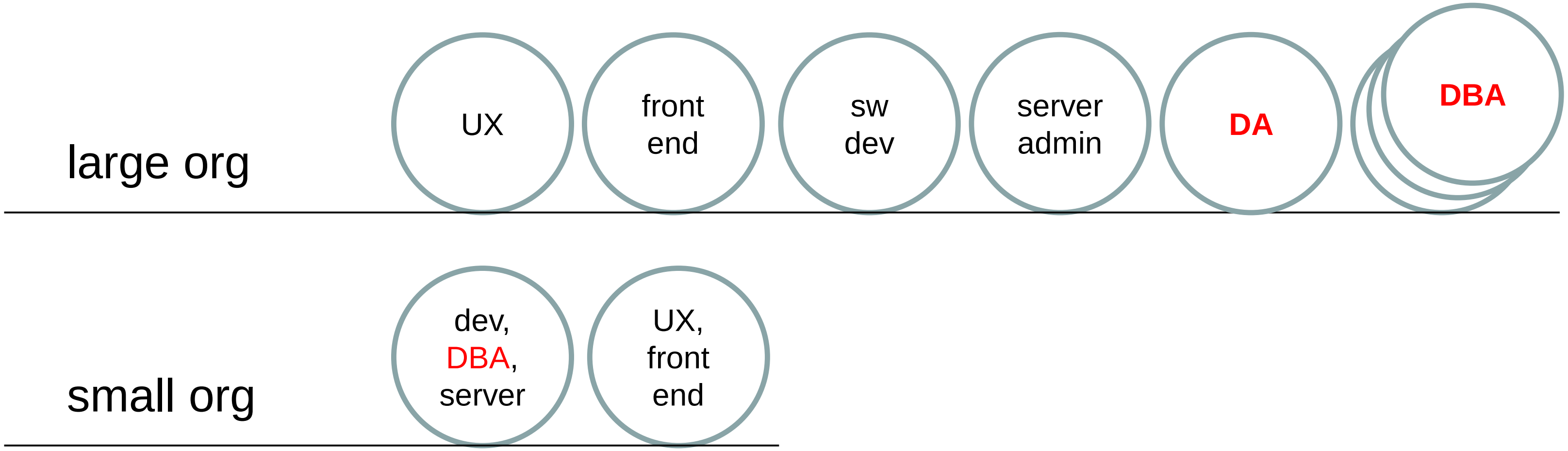
Today's Session…

- The 'Database Administrator' role
- Capacity Planning
  - estimating table growth over time
- Performance
  - storage architecture
  - using indexes to improve performance
  - monitoring performance
- Security
  - threats and responses
  - web apps and SQL injection
- Backup and recovery
  - types of failures, and how to respond
  - types of backups

The DBA role

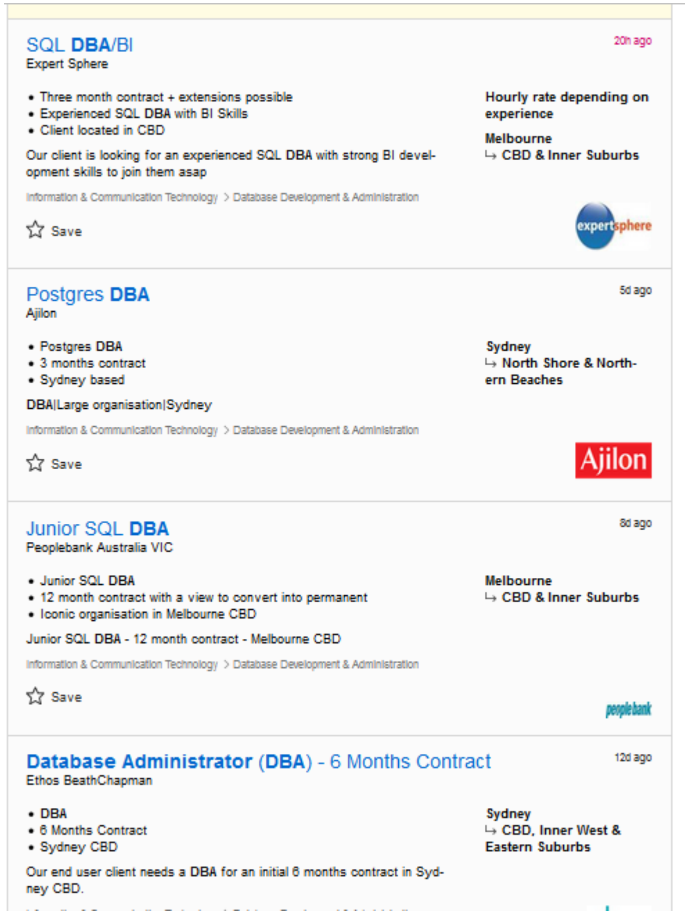
- “Database Administrator” aka “DBA”
- primarily concerned with “maintenance” / “ops” phase
- but should be consulted during design and development
- “person” or “role”?
- large companies may have many DBA’s
- small company – maybe the developer is the DBA
- some DBA tasks are made redundant by *cloud* DBMS





Data and Database Administration

- Data Administrator (management role)
  - data policies, procedures and standards
  - planning
  - data conflict resolution
  - managing info repository (data dictionary)
  - internal marketing
  - similar to “Chief Data Officer”
- Database Administrator (technical role)
  - analyze and design DB
  - select DBMS / tools / vendor
  - install and upgrade DBMS
  - tune DBMS performance
  - manage security, privacy, integrity
  - backup and recovery



SQL **DBA**/BI
Expert Sphere
20h ago

- Three month contract + extensions possible
- Experienced SQL DBA with BI Skills
- Client located in CBD

Our client is looking for an experienced SQL DBA with strong BI development skills to join them asap

Hourly rate depending on experience

Melbourne
↳ CBD & Inner Suburbs

Information & Communication Technology > Database Development & Administration

☆ Save

Postgres **DBA**
Ajilon
5d ago

- Postgres DBA
- 3 months contract
- Sydney based

DBA|Large organisation|Sydney

Sydney
↳ North Shore & Northern Beaches

Information & Communication Technology > Database Development & Administration

☆ Save

Junior SQL **DBA**
Peoplebank Australia VIC
8d ago

- Junior SQL DBA
- 12 month contract with a view to convert into permanent
- Iconic organisation in Melbourne CBD

Junior SQL DBA - 12 month contract - Melbourne CBD

Melbourne
↳ CBD & Inner Suburbs

Information & Communication Technology > Database Development & Administration

☆ Save

Database Administrator (**DBA**) - 6 Months Contract
Ethos BeathChapman
12d ago

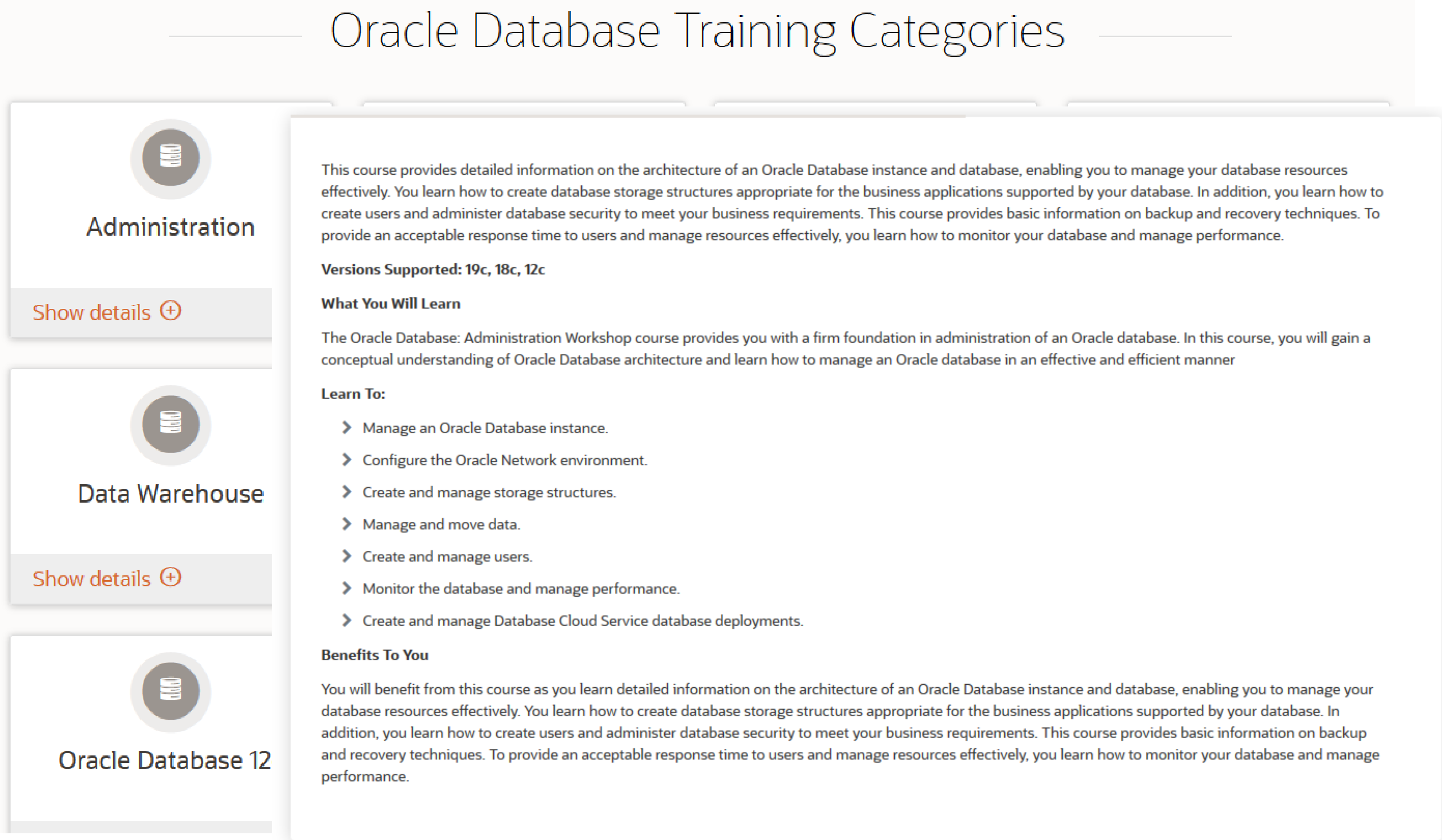
- DBA
- 6 Months Contract
- Sydney CBD

Our end user client needs a DBA for an initial 6 months contract in Sydney CBD.

Sydney
↳ CBD, Inner West & Eastern Suburbs

(Hoffer et al., chapter 11)

DBA certification courses



Oracle Database Training Categories

Administration

Show details ⊕

Data Warehouse

Show details ⊕

Oracle Database 12

This course provides detailed information on the architecture of an Oracle Database instance and database, enabling you to manage your database resources effectively. You learn how to create database storage structures appropriate for the business applications supported by your database. In addition, you learn how to create users and administer database security to meet your business requirements. This course provides basic information on backup and recovery techniques. To provide an acceptable response time to users and manage resources effectively, you learn how to monitor your database and manage performance.

Versions Supported: 19c, 18c, 12c

What You Will Learn

The Oracle Database: Administration Workshop course provides you with a firm foundation in administration of an Oracle database. In this course, you will gain a conceptual understanding of Oracle Database architecture and learn how to manage an Oracle database in an effective and efficient manner

Learn To:

- Manage an Oracle Database instance.
- Configure the Oracle Network environment.
- Create and manage storage structures.
- Manage and move data.
- Create and manage users.
- Monitor the database and manage performance.
- Create and manage Database Cloud Service database deployments.

Benefits To You

You will benefit from this course as you learn detailed information on the architecture of an Oracle Database instance and database, enabling you to manage your database resources effectively. You learn how to create database storage structures appropriate for the business applications supported by your database. In addition, you learn how to create users and administer database security to meet your business requirements. This course provides basic information on backup and recovery techniques. To provide an acceptable response time to users and manage resources effectively, you learn how to monitor your database and manage performance.

THE UNIVERSITY OF MELBOURNE

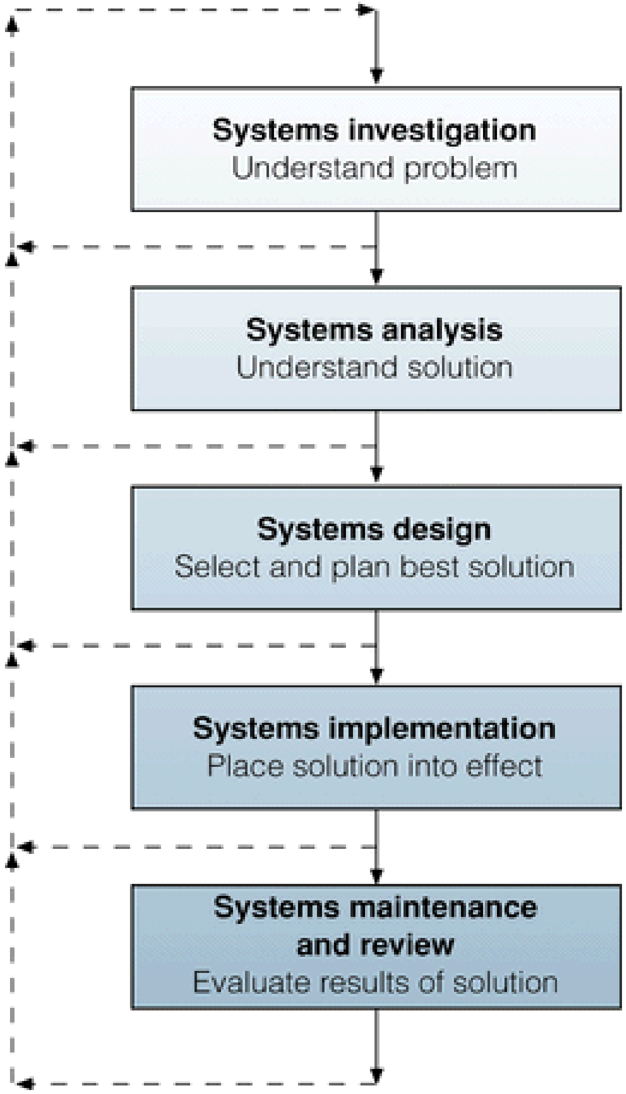
Capacity Planning

What is 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 (we will focus on this)
  - transaction throughput
  - (at go-live and throughout the life of the system)

Capacity Planning in the dev life cycle



Systems investigation
Understand problem

Systems analysis
Understand solution

Systems design
Select and plan best solution

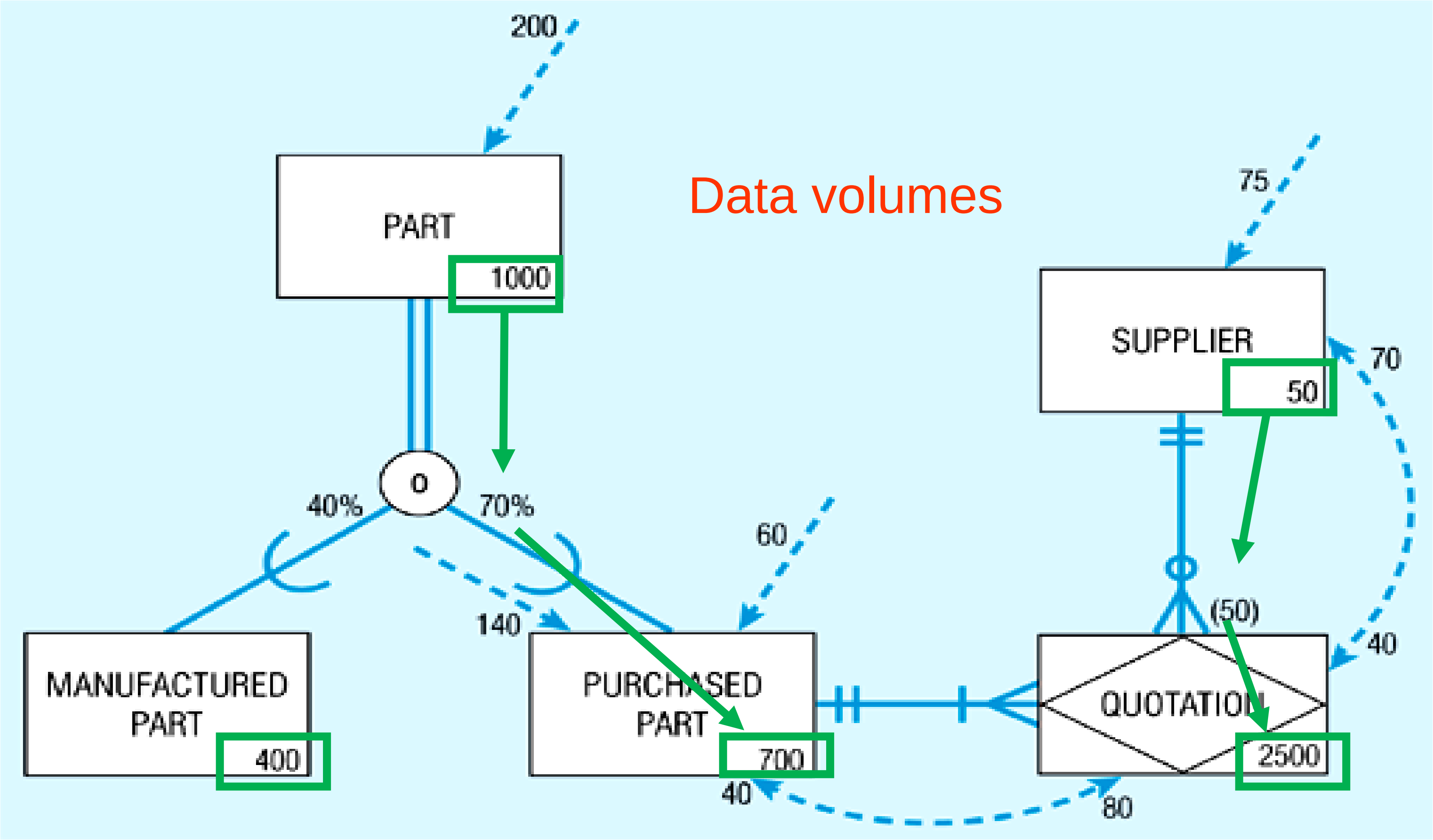
Systems implementation
Place solution into effect

Systems maintenance and review
Evaluate results of solution

calculate expected disk usage and transaction load

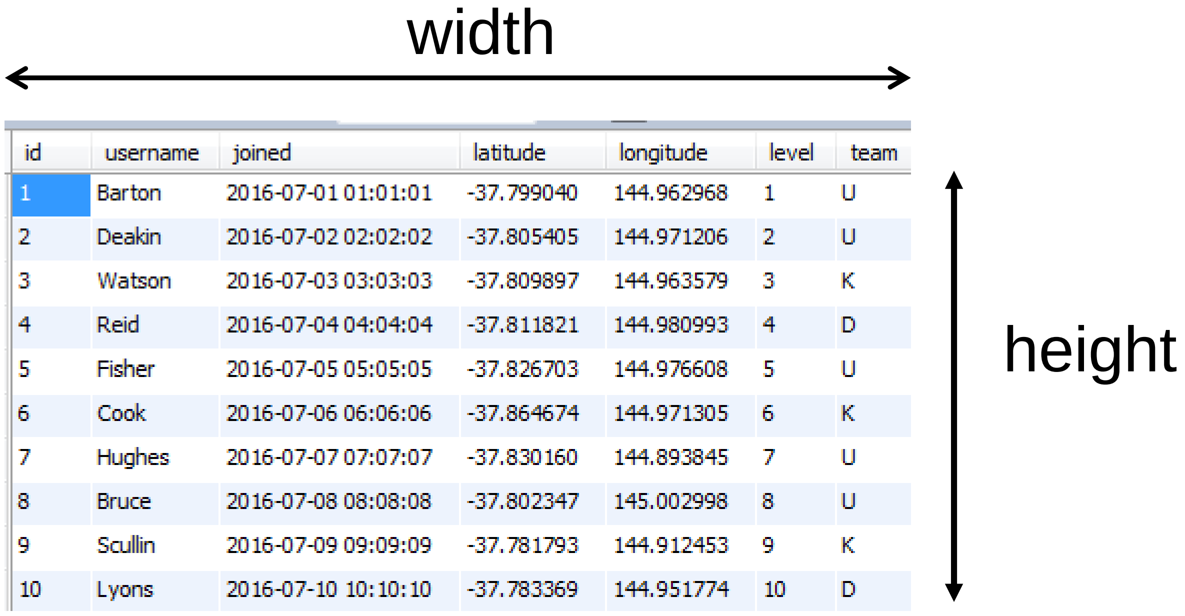
monitor and predict disk usage and transaction load

Estimating Database Usage



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*
  - where table size = number of rows * average row width



width

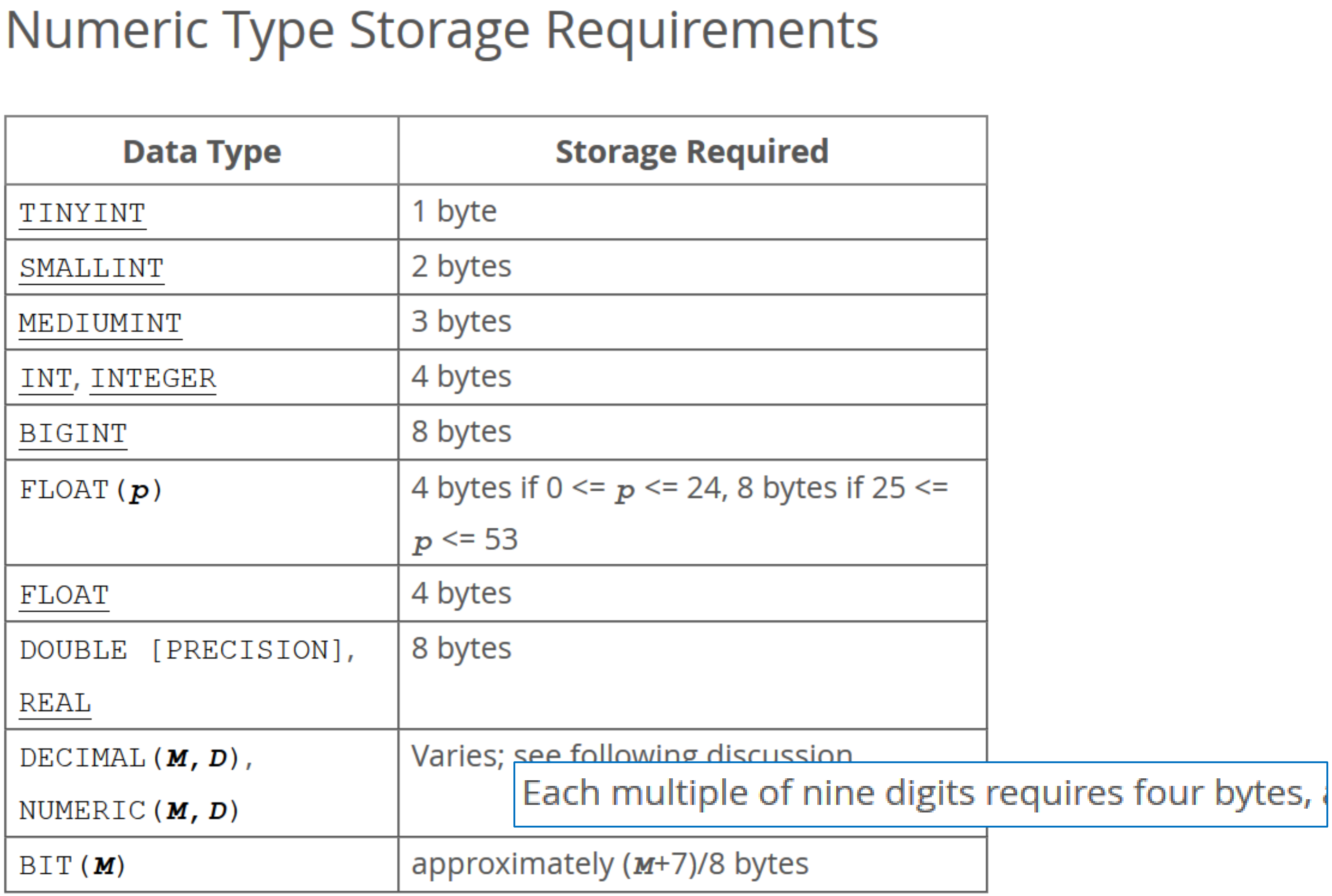
id	username	joined	latitude	longitude	level	team
1	Barton	2016-07-01 01:01:01	-37.799040	144.962968	1	U
2	Deakin	2016-07-02 02:02:02	-37.805405	144.971206	2	U
3	Watson	2016-07-03 03:03:03	-37.809897	144.963579	3	K
4	Reid	2016-07-04 04:04:04	-37.811821	144.980993	4	D
5	Fisher	2016-07-05 05:05:05	-37.826703	144.976608	5	U
6	Cook	2016-07-06 06:06:06	-37.864674	144.971305	6	K
7	Hughes	2016-07-07 07:07:07	-37.830160	144.893845	7	U
8	Bruce	2016-07-08 08:08:08	-37.802347	145.002998	8	U
9	Scullin	2016-07-09 09:09:09	-37.781793	144.912453	9	K
10	Lyons	2016-07-10 10:10:10	-37.783369	144.951774	10	D

height

THE UNIVERSITY OF MELBOURNE

Estimating row widths: numbers

- need to know storage size of different data types
- https://dev.mysql.com/doc/refman/8.0/en/storage-requirements.html



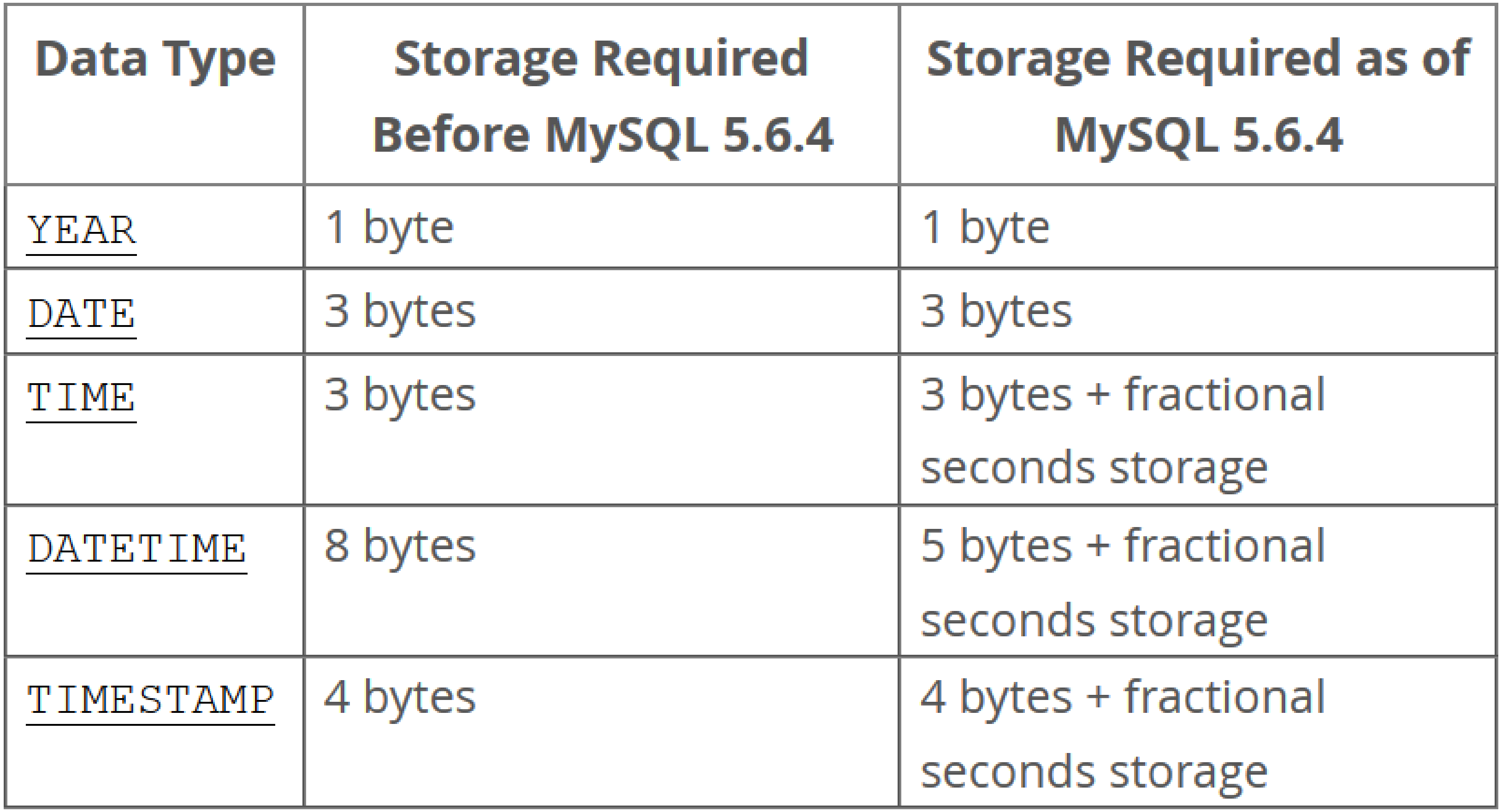
Numeric Type Storage Requirements

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

Each multiple of nine digits requires four bytes,

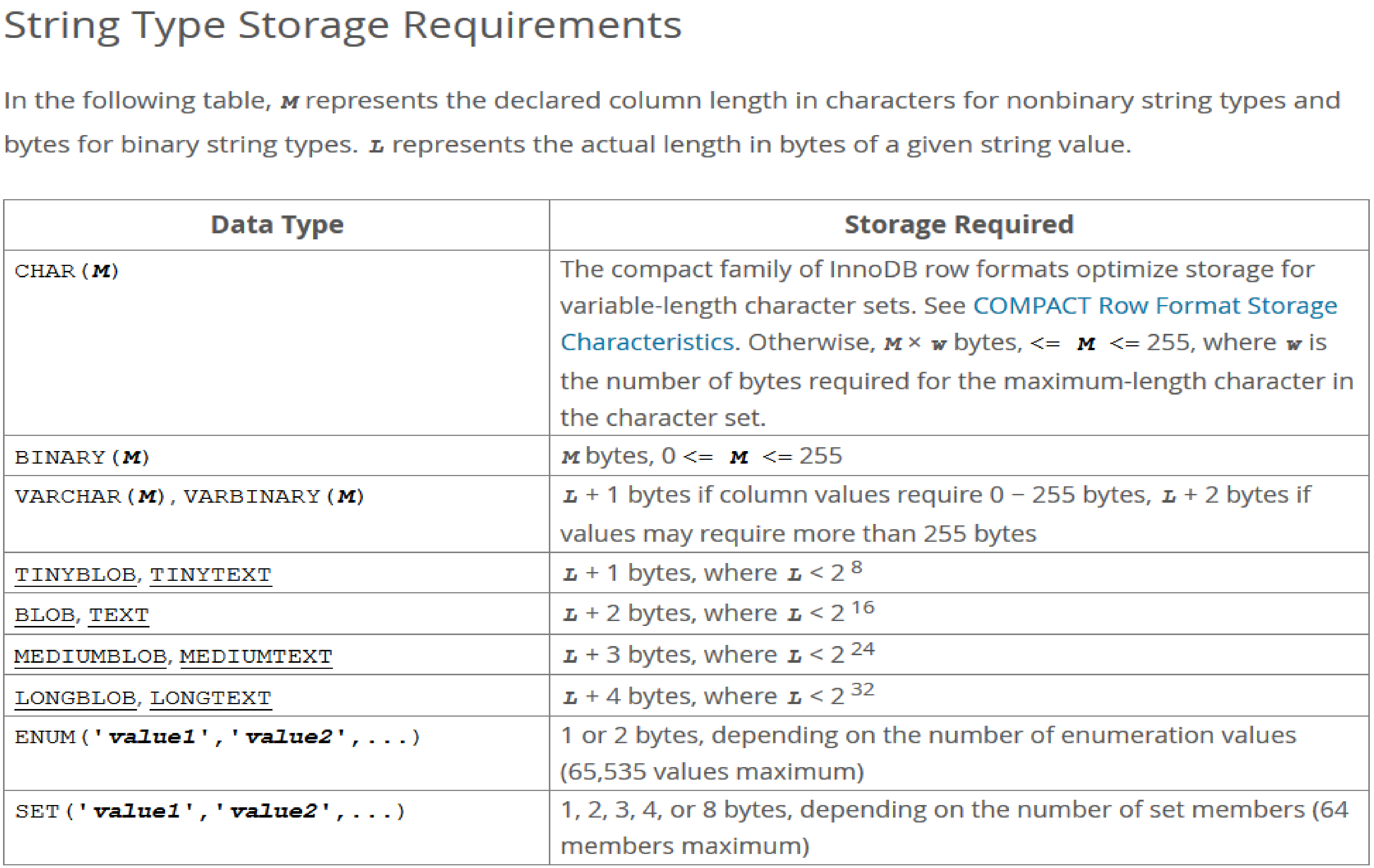
Estimating row widths: dates and times

- (these sizes are for MySQL and are slightly different for other vendors)



Data Type	Storage Required Before MySQL 5.6.4	Storage Required as of MySQL 5.6.4
YEAR	1 byte	1 byte
DATE	3 bytes	3 bytes
TIME	3 bytes	3 bytes + fractional seconds storage
DATETIME	8 bytes	5 bytes + fractional seconds storage
TIMESTAMP	4 bytes	4 bytes + fractional seconds storage

Estimating row widths: text



String Type Storage Requirements

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)`	The compact family of InnoDB row formats optimize storage for variable-length character sets. See COMPACT Row Format Storage Characteristics. Otherwise, M × w bytes, $<= M <= 255$, where w is the number of bytes required for the maximum-length character in the character set.
`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)
`SET('value1','value2',...)`	1, 2, 3, 4, or 8 bytes, depending on the number of set members (64 members maximum)

Estimate growth of tables

For example: Using this simple database in which users post to forums, assume there are:

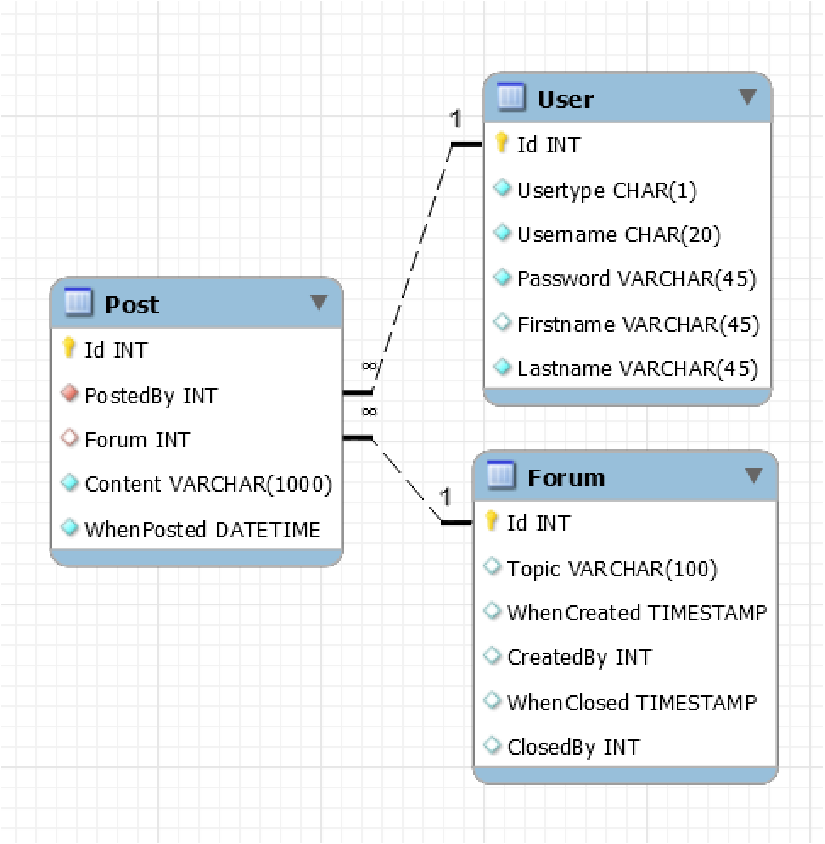
- 100 forums
- 1 million users

and assume that:

- users post on average 30 times per month

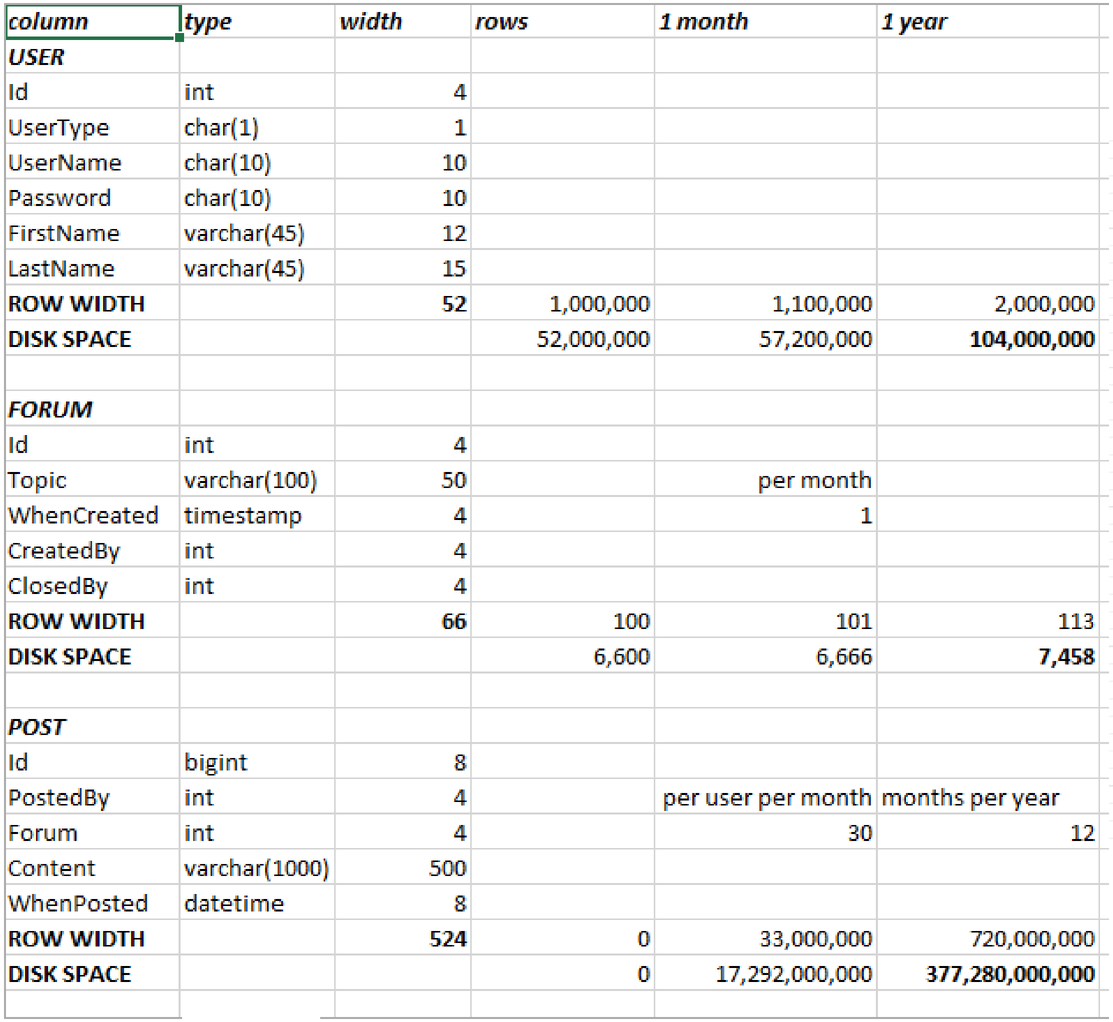
we calculate:

- *Post* table grows by 1M rows / day
- which means 12 Inserts per second

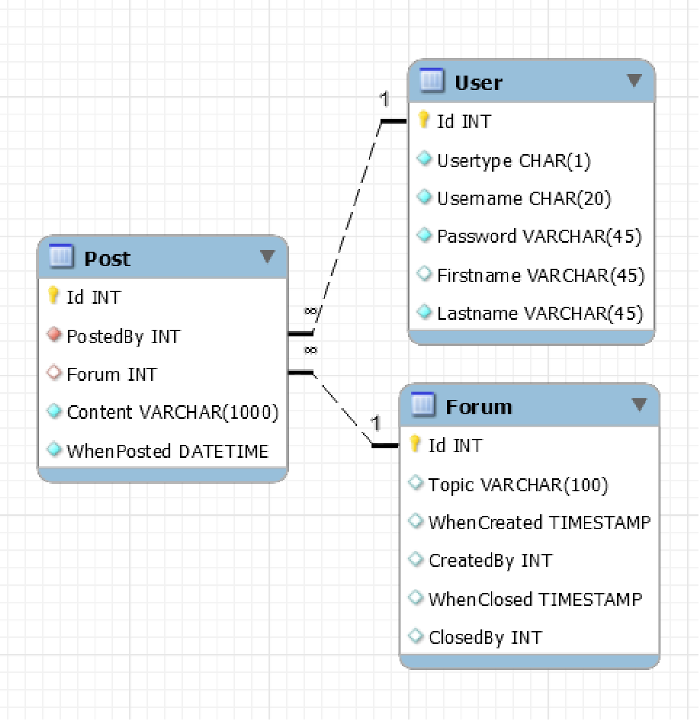


Calculate disk space per table

- use a spreadsheet to simplify calculations and enable what-ifs

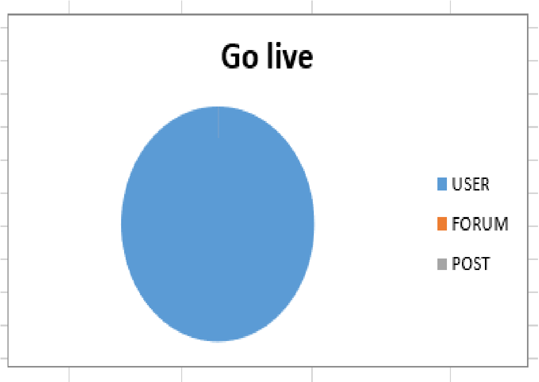


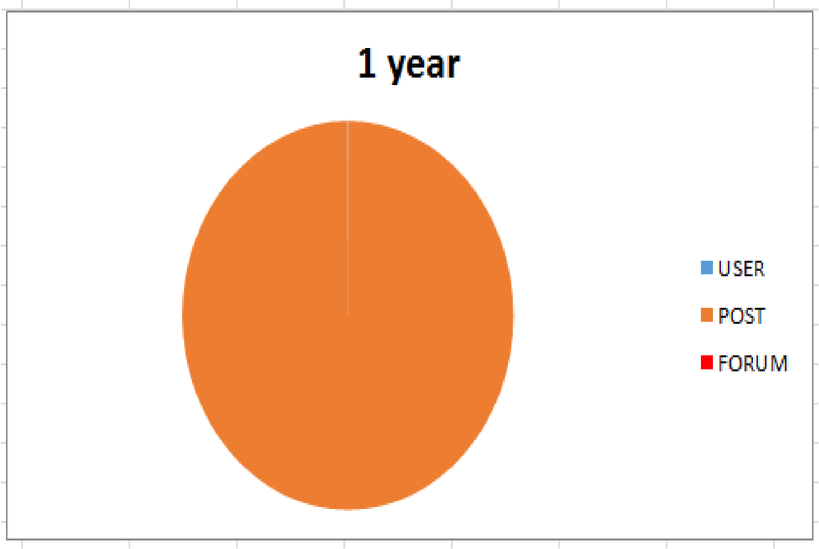
column	type	width	rows	1 month	1 year
USER					
Id	int	4			
UserType	char(1)	1			
UserName	char(10)	10			
Password	char(10)	10			
FirstName	varchar(45)	12			
LastName	varchar(45)	15			
ROW WIDTH		52	1,000,000	1,100,000	2,000,000
DISK SPACE			52,000,000	57,200,000	104,000,000
FORUM					
Id	int	4			
Topic	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	months per year
Forum	int	4		30	12
Content	varchar(1000)	500			
WhenPosted	datetime	8			
ROW WIDTH		524	0	33,000,000	720,000,000
DISK SPACE			0	17,292,000,000	377,280,000,000



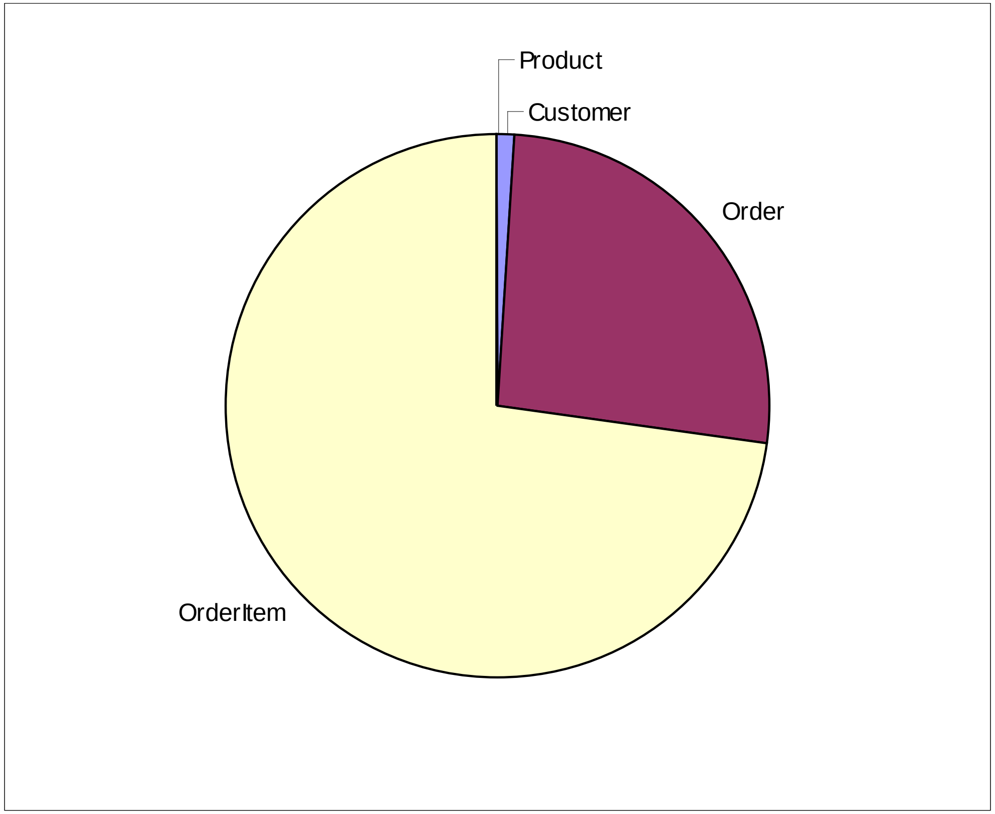
Projected total storage requirements

Table	*Row width*	*No. rows at 1 year*	*Disk space at 1 year*
User	52 bytes	2,000,000	104 Mb
Forum	66 bytes	113	0.007 Mb
Post	524 bytes	720 Mb	377 Gb





Growth of different kinds of tables



In OLTP databases, “event” tables typically grow much faster than “entity” tables.

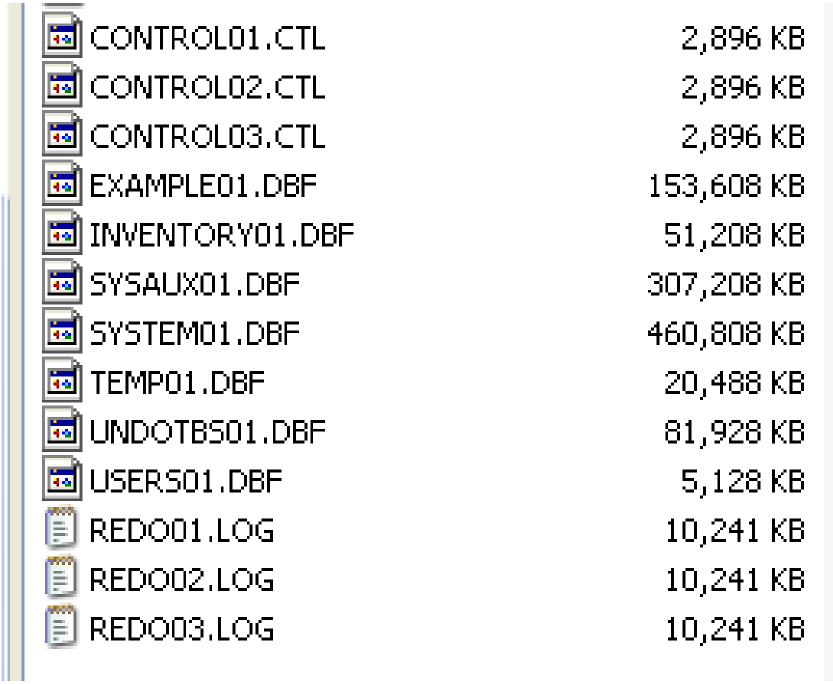
Be aware of which tables are the biggest consumers of space.

Some DBMS allow placing high-volume tables on separate disks.

Other disk space requirements

- Inside tables, in addition to row data,
  there is unused space at the data file and block level

- The DBMS also needs space for other files
  - for example (Oracle)
  - control file(s),
  - data dictionary,
  - indexes,
  - undo area,
  - sort area,
  - redo logs



CONTROL01.CTL	2,896 KB
CONTROL02.CTL	2,896 KB
CONTROL03.CTL	2,896 KB
EXAMPLE01.DBF	153,608 KB
INVENTORY01.DBF	51,208 KB
SYSAUX01.DBF	307,208 KB
SYSTEM01.DBF	460,808 KB
TEMP01.DBF	20,488 KB
UNDOTBS01.DBF	81,928 KB
USERS01.DBF	5,128 KB
REDO01.LOG	10,241 KB
REDO02.LOG	10,241 KB
REDO03.LOG	10,241 KB

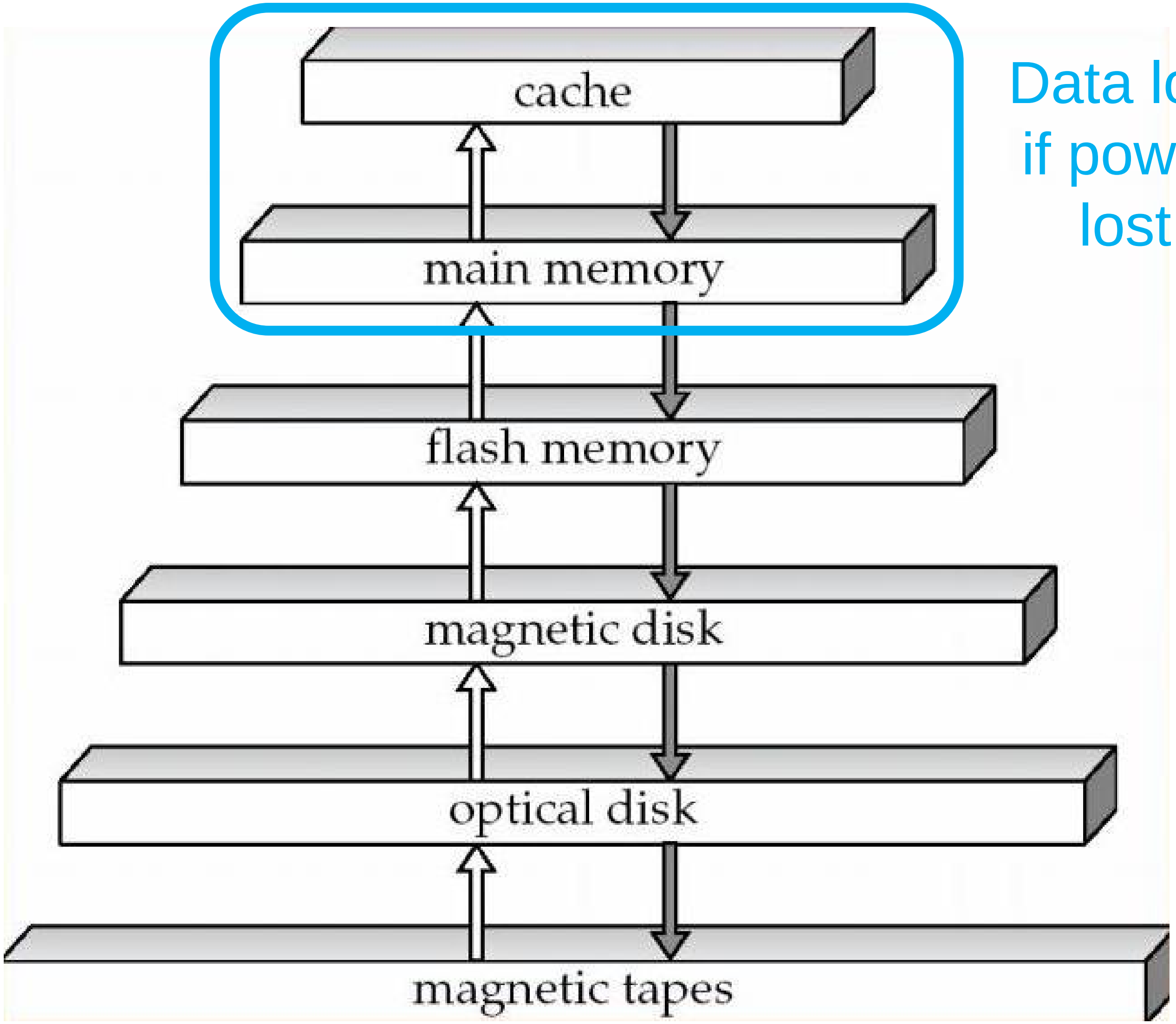
Database Performance

THE UNIVERSITY OF MELBOURNE

Storage media hierarchy

faster, more expensive, smaller capacity

slower, cheaper, *older*, bigger capacity



Data lost if power lost

What affects database performance?

- caching data in memory, e.g. data buffers
- placement of data files across disc drives
- database replication and server clustering
- use of fast storage such as SSD
- use of indexes to speed up searches and joins
- good choice of data types
- good program logic
- good query execution plans
- good application code (e.g. no deadlocks)

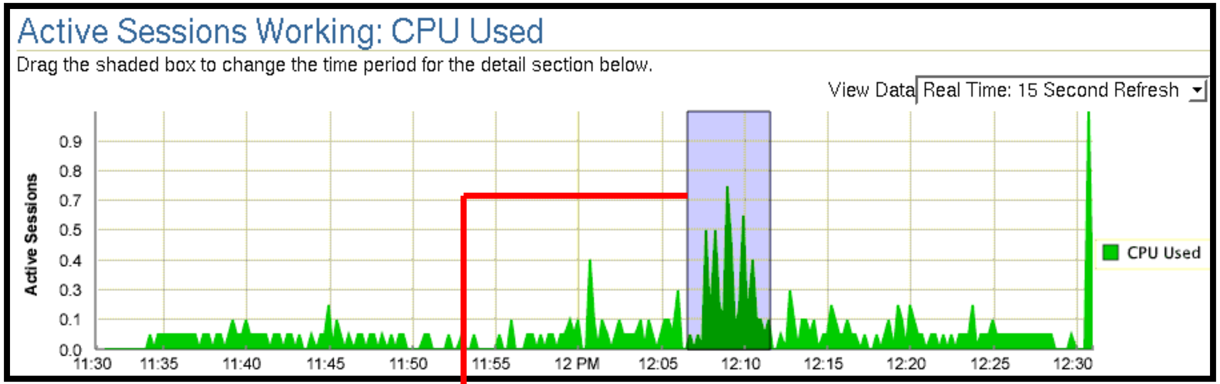
When to create indexes

source: Oracle® Database Application Developer's Guide

For each table, choose the columns you will index, based on:

- if the column is *queried frequently* (used in Where clauses)
- columns that are used for *joins*
- primary *keys* (automatic in most DBMS)
- foreign *keys* (automatic in MySQL)
- unique columns (automatic in most DBMS)
- large tables only - small tables do not require indexes
- columns where values don’t change too often
- if you frequently retrieve less than about 15% of the rows
- wide range of values (good for regular indexes).
- small range of values (good for bitmap indexes).

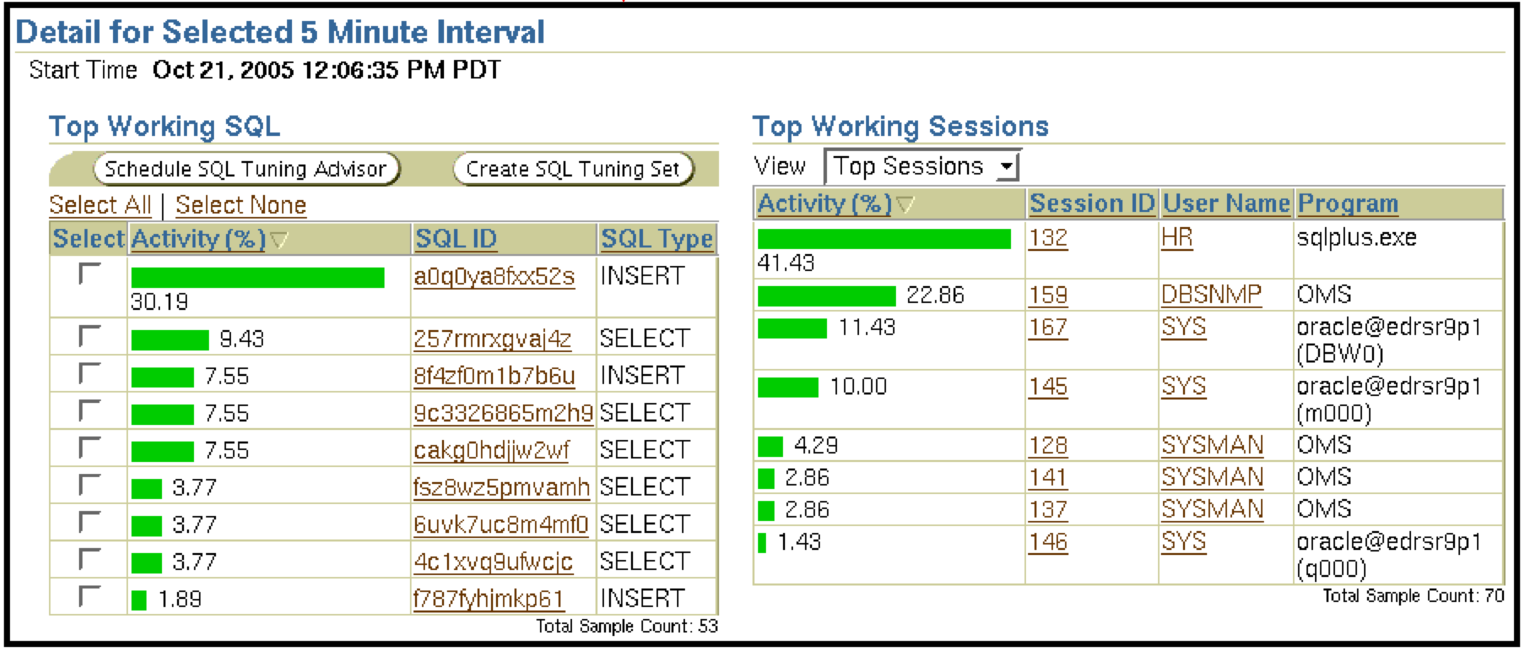
Performance Monitoring



Active Sessions Working: CPU Used

Drag the shaded box to change the time period for the detail section below.

View Data Real Time: 15 Second Refresh



Detail for Selected 5 Minute Interval

Start Time **Oct 21, 2005 12:06:35 PM PDT**

Top Working SQL

Schedule SQL Tuning Advisor | Create SQL Tuning Set

Select All | Select None

Select	Activity (%) ▽	SQL ID	SQL Type
☐	30.19	a0q0ya8fxx52s	INSERT
☐	9.43	257rmrxgvaj4z	SELECT
☐	7.55	8f4zf0m1b7b6u	INSERT
☐	7.55	9c3326865m2h9	SELECT
☐	7.55	cakg0hdjjw2wf	SELECT
☐	3.77	fsz8wz5pmvamh	SELECT
☐	3.77	6uvk7uc8m4mf0	SELECT
☐	3.77	4c1xvq9ufwcjc	SELECT
☐	1.89	f787fyhjmkp61	INSERT

Total Sample Count: 53

Top Working Sessions

View Top Sessions

Activity (%) ▽	Session ID	User Name	Program
41.43	132	HR	sqlplus.exe
22.86	159	DBSNMP	OMS
11.43	167	SYS	oracle@edrsr9p1 (DBW0)
10.00	145	SYS	oracle@edrsr9p1 (m000)
4.29	128	SYSMAN	OMS
2.86	141	SYSMAN	OMS
2.86	137	SYSMAN	OMS
1.43	146	SYS	oracle@edrsr9p1 (q000)

Total Sample Count: 70

Security

Database Security

- Database security covers a number of areas
  - legal and ethical issues
  - policy issues
  - system-related issues
  - need to identify multiple security levels

Threats to databases

- Loss of integrity
  - keep data consistent
  - free of errors or anomalies
- Loss of availability
  - want database to be available to users
- Loss of confidentiality
  - must be protected against unauthorized access
- To protect databases against these types of threats, different kinds of countermeasures can be implemented:
  - access control
  - encryption

Access Control

- The security mechanism of a DBMS must include provisions for restricting access to data
- Access control is handled by the DBA creating user accounts for those with a legitimate need to access the DB
- The database keeps track of all operations on the database for all users (usage log)
- When tampering is suspected, perform an audit
  - a database audit consists of reviewing the log to examine all accesses and operations applied to the database during a certain time period
- Need to control online *and physical* access to the database

Access Control

- Based on granting and revoking privileges
- Types of discretionary privileges
  - account level
    - DBA specifies the particular privileges that each user holds regarding the database as a whole, i.e. the operations they can carry out on the database
  - table level
    - DBA controls a user's privilege to access particular tables or views
  - schema level
    - DBA controls a user's privilege to access a particular schema in the database.

see the list of MySQL user privileges at

http://dev.mysql.com/doc/refman/5.7/en/grant.html

Access control using Views

- Views are an important discretionary authorization mechanism
- Views are good for
  - hiding the database structure
  - hiding some data (ie columns in tables)
- for example
  - if the owner A of a table T wants another user B to be able to retrieve only some columns of T, A can create a view V of T that includes only those columns and then grant SELECT on V to B.
  - to limit B to retrieving only certain rows of T, a view V’ can be created that selects only those rows from T

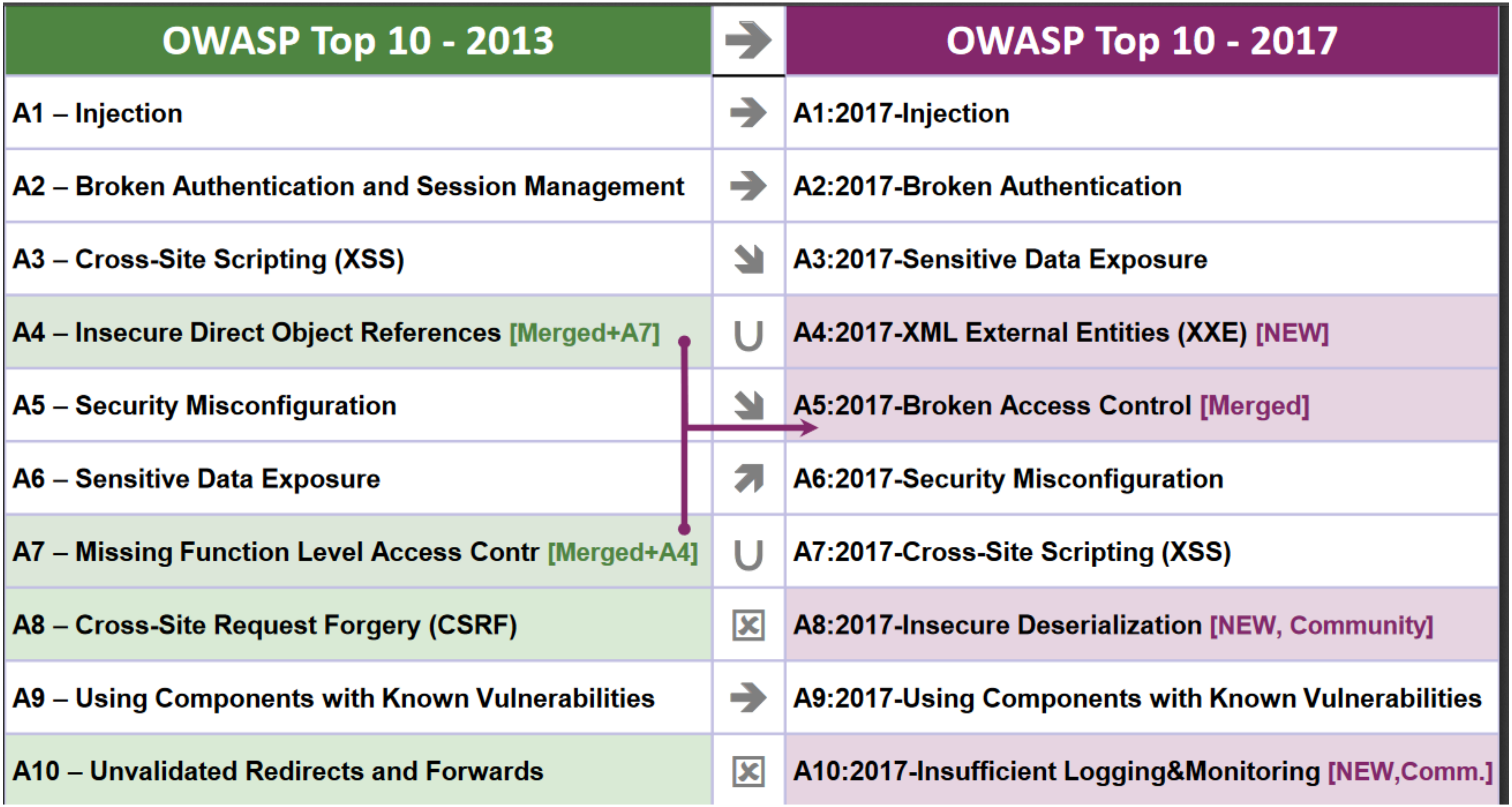
Encryption

- Particular tables or columns may be encrypted to:
  - protect sensitive data (e.g. password) when they are transmitted over a network
    - prevents interception by third party
  - encrypt data in the database (e.g. credit card numbers)
    - provides some protection in case of unauthorized access

- Data is encoded using an algorithm
  - authorized users are given keys to decipher data

OWASP top ten web app vulnerabilities

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

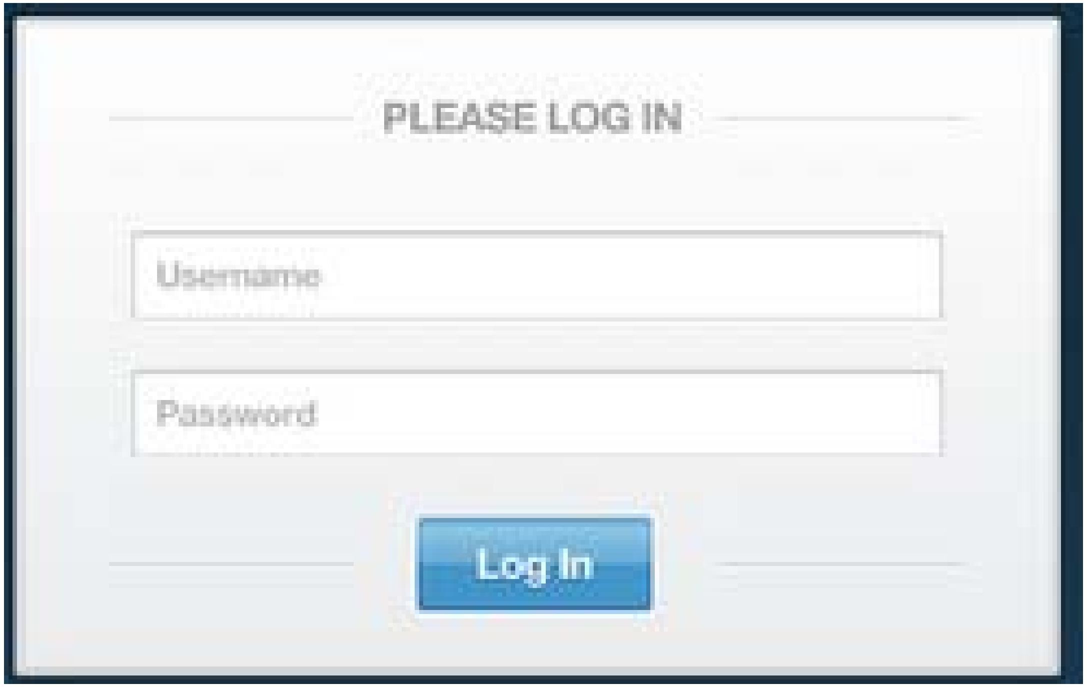


OWASP Top 10 - 2013	→	OWASP Top 10 - 2017
A1 – Injection	→	A1:2017-Injection
A2 – Broken Authentication and Session Management	→	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	↘	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	∪	A4:2017-XML External Entities (XXE) [NEW]
A5 – Security Misconfiguration	↘	A5:2017-Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6:2017-Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	∪	A7:2017-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	☒	A8:2017-Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9:2017-Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	☒	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

Protecting against SQL injection

- SQL Injection attacks
  - a technique used to exploit web applications that use *user input within database queries*
  - malicious code is entered into a data entry field to manipulate SQL commands that are run against the database
  - How to prevent:
    - sanitize user inputs
    - pass inputs as parameters to a stored procedure, rather than directly building the SQL string in the code

SQL injection example: login


PLEASE LOG IN

Username

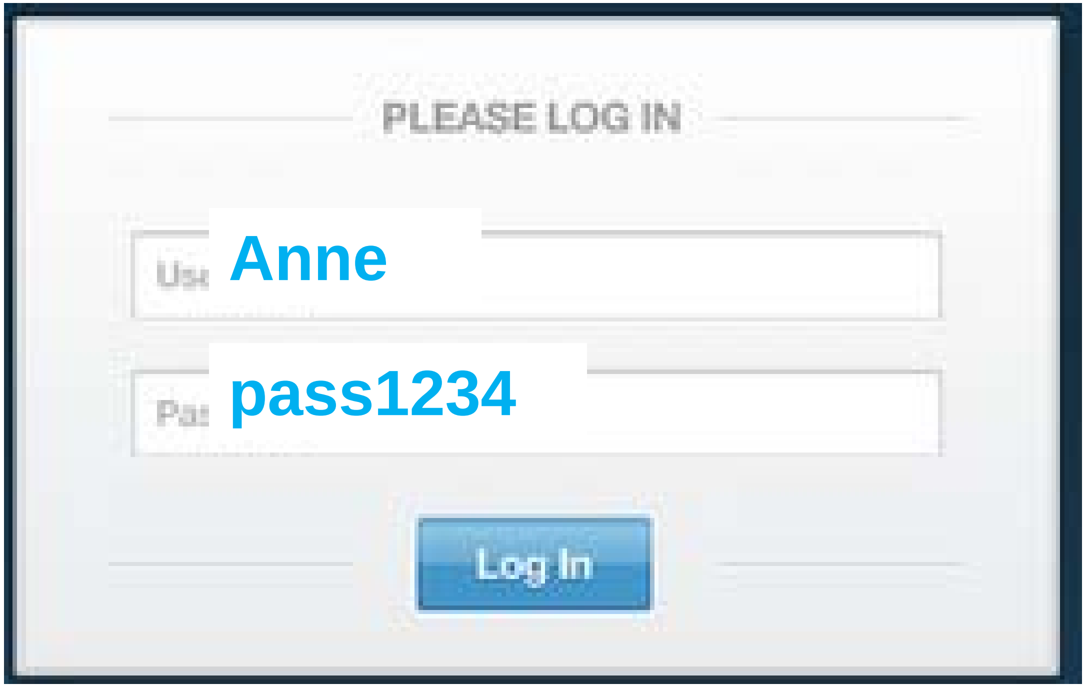
Password

Log In

- user inputs are used to form an SQL statement
- statement is executed

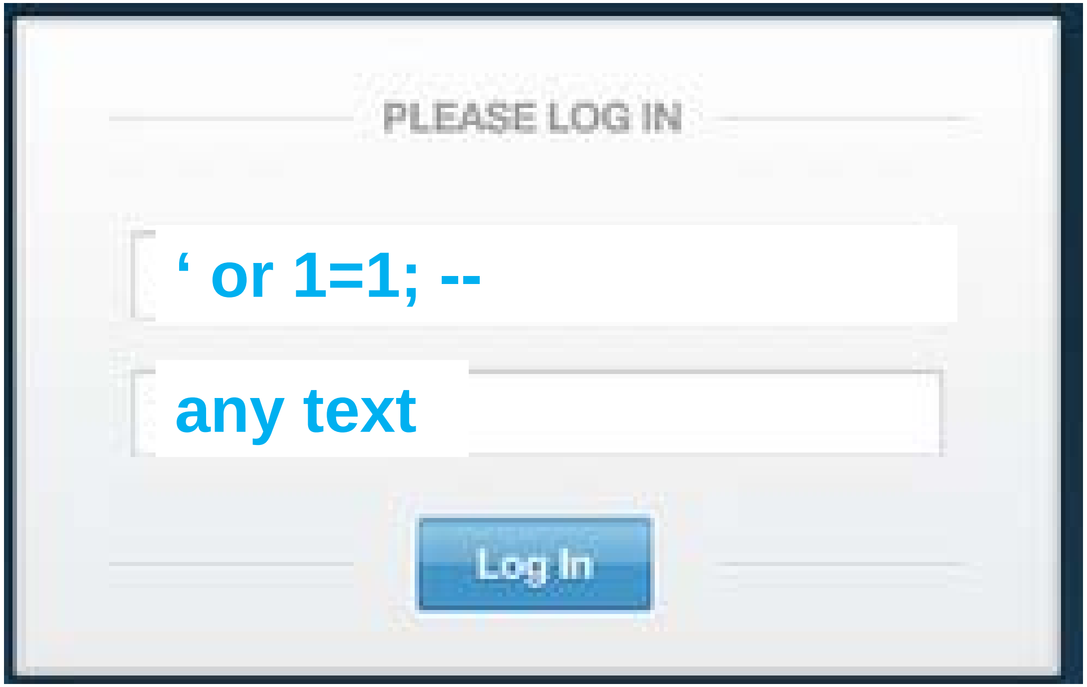
```
select * from User
where username = ' @name '
and password = ' @pw ';
```

SQL injection: what programmer wants



```
select * from User
where username = 'Anne'
and password = 'pass1234';
```

SQL injection: malicious input



‘ or 1=1; --

any text

text entered in @name string now

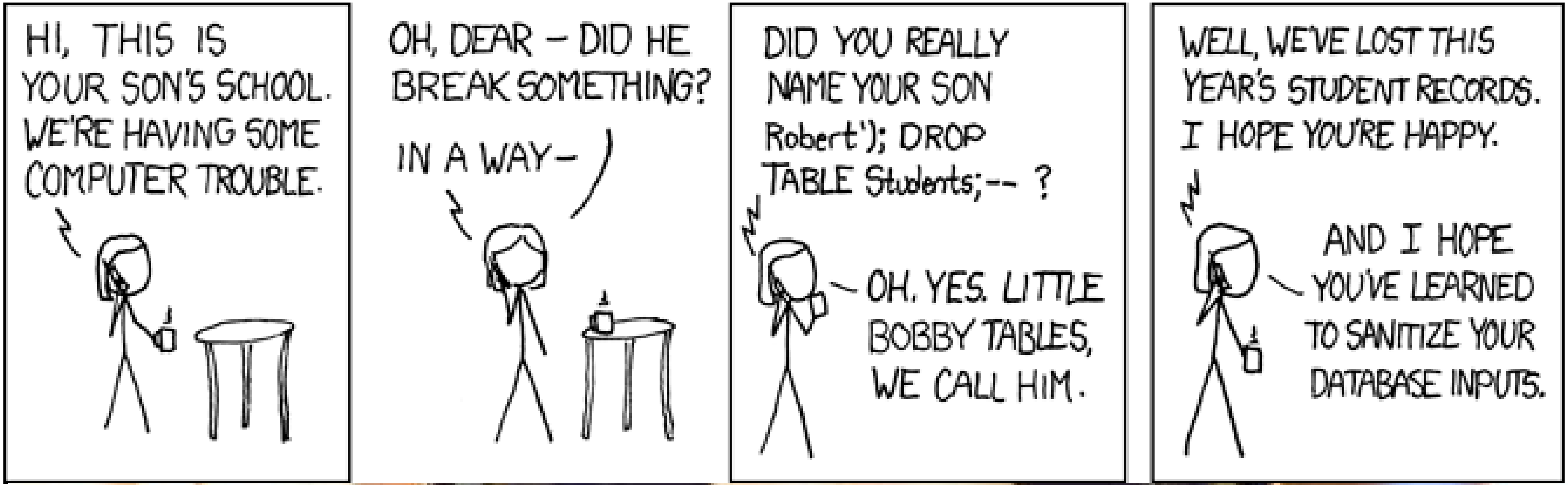
- closes the string
- adds a condition that is always true
- ends the SQL statement
- begins a comment with '– ' to neutralize the rest of the SQL

```
select * from User
where username = '' or 1=1; --  '
and password = 'any text';
```

SQL injection: prevention

- Primary defences:
  - **Prepared Statements (parameterized queries)**
  - **Stored Procedures**
    - (both mean SQL is no longer 'dynamic')
  - **“Escape” all user input**
    - turns SQL special characters like ‘ ; -- into ordinary characters
- Additional defences:
  - **Principle of Least Privilege**
    - don't give application accounts DBA privileges
  - **White List input validation**
    - check input is from a list of acceptable values
      - Source: OWASP SQL Injection Prevention Cheat Sheet

SQL Injection in popular culture


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

Backup and Recovery

Types of database failure

Backup your data

Test your backups

Recover your data

What is a Backup

- A backup is an extra copy of your data
  - there are several types of backup
- If data becomes corrupted or deleted,
  it can be restored from the backup copy
- A backup and recovery strategy is needed
  - to plan in advance how data is backed up
  - to plan in advance 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 drives
  - CPU
  - memory

Must also protect against

- malicious activity
  - security compromise
    - server, database, application

e.g: Police lose 8 years of criminal evidence

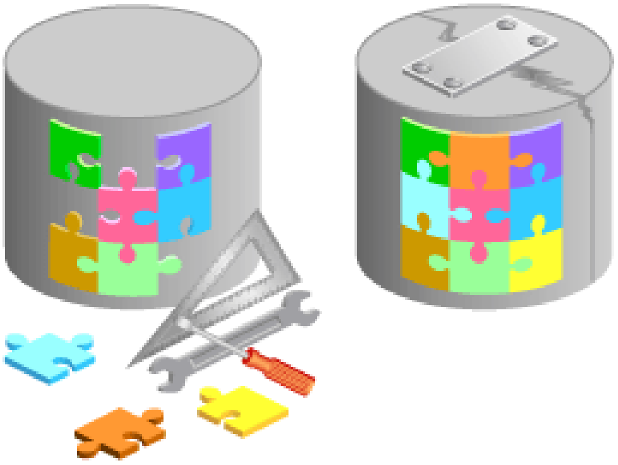
- natural or man made disasters
  - consider the *scale* of the damage

- government regulation
  - historical archiving rules
  - Metadata collection (Australia)
  - HIPPA, EU data retention regulations

Categories of Failures

Failures can generally be divided into the following categories:

- Statement failure
- User process failure
- Network failure
- User error
- Instance failure
- Media failure



Types of Backups

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

Physical vs Logical backup

- Physical backup
  - “raw” copies of files and directories
  - suitable for large databases that need fast recovery
  - database is preferably offline during backup (“cold” backup)
  - 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

Physical vs Logical backup

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

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

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
- Incremental
  - only the changes since last backup are backed up
  - to restore:
    - restore last full backup
    - then restore incrementals since that time

Create a Backup Policy

- Backup strategy is usually a combination of full and incremental backups
  - for example:
    - weekly full backup
    - daily incremental backup
- Conduct backups when database load is low
- if *replicated* database, use the mirror database for backups to negate any performance concerns with the main database
- TEST your backup before you NEED your backup!

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)

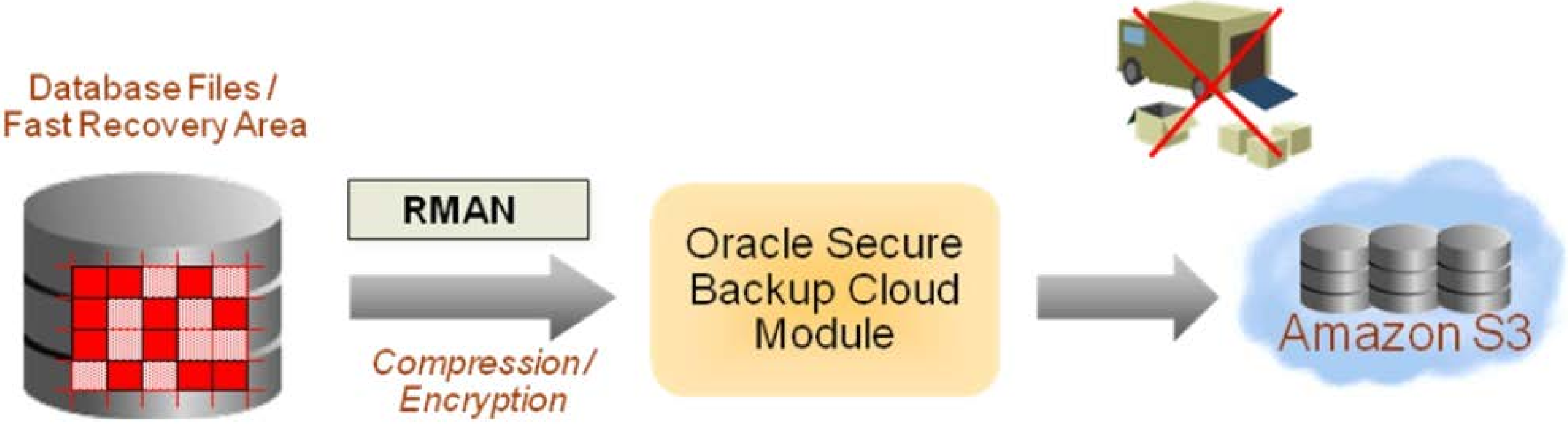

Database Files / Fast Recovery Area
RMAN
Compression/ Encryption
Oracle Secure Backup Cloud Module
Amazon S3

Figure 1. Oracle Database backup in the Cloud