MONASH University

MONASH
INFORMATION
TECHNOLOGY

DB Connectivity
Web Technology
Cloud Computing

GROUP OF EIGHT AUSTRALIA

Where Are We

- Through this unit we have looked at:
  - The fundamental principles on which relational databases are built
  - Designing a relational database and
  - Implementing a relational database and manipulating its data via SQL

- In practice the database you create & populate will be used by *normal users* not database professionals
  - set of tables/views created under one account
  - control access to this accounts objects

Database connectivity

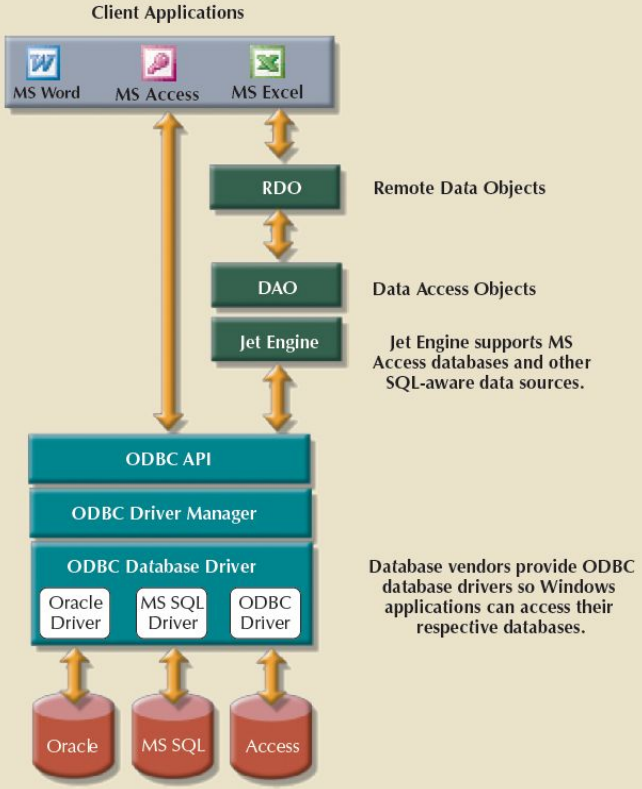
Q1. Which of the following are considered as application layer (multiple answers are possible):

a. Moodle
b. WES
c. Allocate+
d. Ms Word

Database Connectivity

- The DATA LAYER – your data management application (DBMS)
- The DATABASE MIDDLEWARE – manages connectivity and data transformation issues. Many options available such as:
  - –Native SQL Connectivity
    - •Vendor provided eg. Oracle SQL*Net
  - – Microsoft ODBC, DAO, RDO; OLE-DB and ADO.NET
  - – Java Database Connectivity (JDBC)
- The APPLICATION – the external interface, mostly in the form of an Application Programming Interface (API)

FIGURE 15.2 USING ODBC, DAO, AND RDO TO ACCESS DATABASES



Coronel & Morris
Fig 15.2 Ed 13

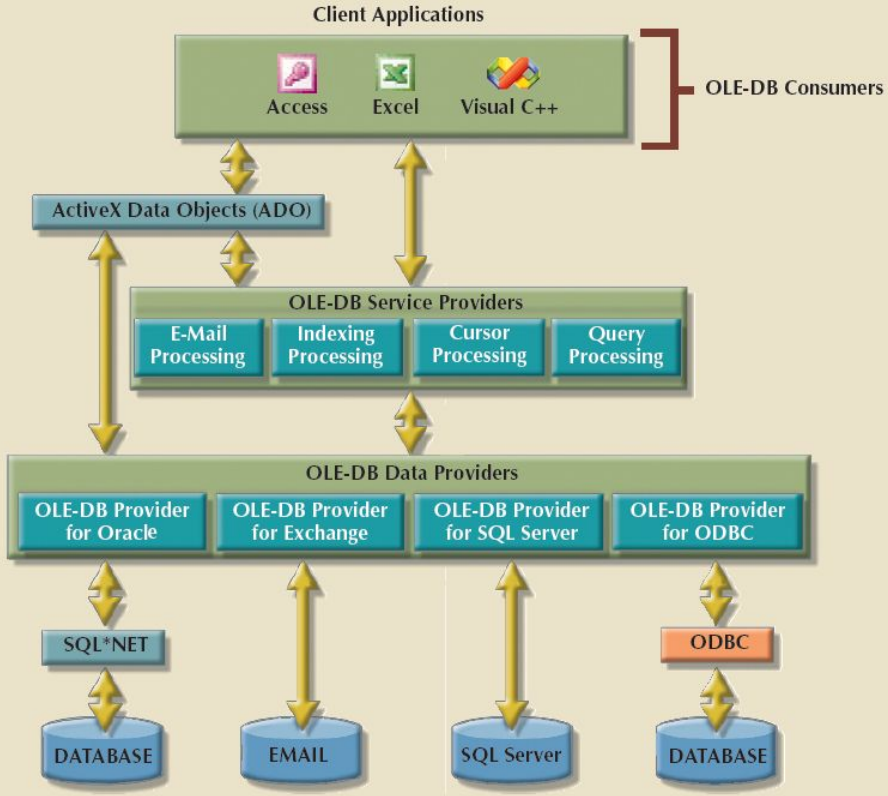
iODBC.org
Independent Open DataBase Connectivity for Linux, MacOS X and Unix systems

MONASH University

Q2. Which of the following database middleware support access to non-relational database:

a. JDBC
b. DAO
c. ODBC
d. OLE-DB

FIGURE 15.5 OLE-DB ARCHITECTURE

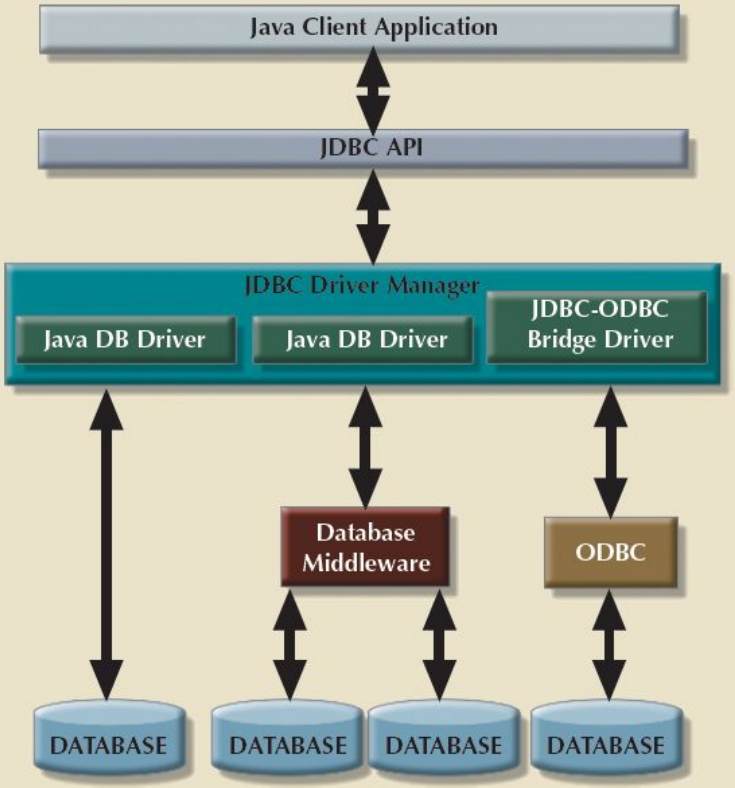


Coronel & Morris
Fig 15.5 Ed 13

Q3. Your team is about to develop a Java based application that communicates with the Oracle server, pick a suitable database middleware in this scenario:

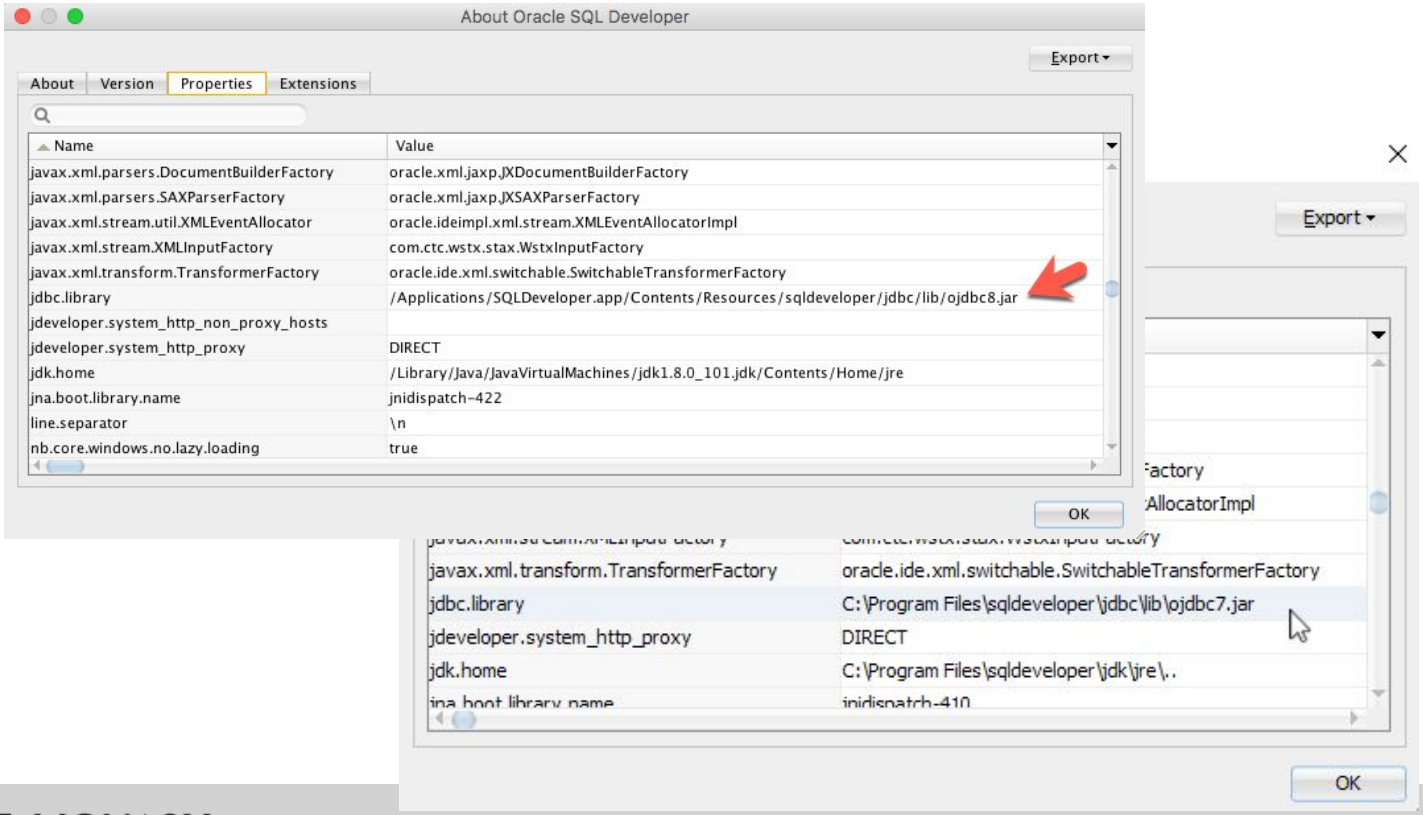
a. JDBC
b. RDO
c. ODBC
d. OLE-DB

FIGURE 15.7 JDBC ARCHITECTURE



Coronel & Morris
Fig 15.7 Ed 13

SQLDeveloper - JDBC



Sample JDBC code snippet

```
public static void viewTable(Connection con, String dbName)
    throws SQLException {

    Statement stmt = null;
    String query = "select COF_NAME, SUP_ID, PRICE, " +
                   "SALES, TOTAL " +
                   "from " + dbName + ".COFFEES";
    try {
        stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery(query);
        while (rs.next()) {
            String coffeeName = rs.getString("COF_NAME");
            int supplierID = rs.getInt("SUP_ID");
            float price = rs.getFloat("PRICE");
            int sales = rs.getInt("SALES");
            int total = rs.getInt("TOTAL");
            System.out.println(coffeeName + "\t" + supplierID +
                               "\t" + price + "\t" + sales +
                               "\t" + total);
        }
    } catch (SQLException e ) {
        JDBCTutorialUtilities.printSQLException(e);
    } finally {
        if (stmt != null) { stmt.close(); }
    }
}
```

Oracle JDBC Tutorial
https://goo.gl/p1bl2b

Oracle Python Tutorial
https://www.oracletutorial.com/python-oracle/

Q4. What is the best practice of accessing data in the database from an application:

a. Write the SQL statements (insert, update, delete) to access the tables as part of the application code
b. Create stored procedures in the DBMS then access and call them from the application
c. It does not matter, as long as it shows the correct output to the users

Placing application logic in the backend

- In this approach we code database objects which "black box" the logic and store them in the database
- Procedures and Packages
  - written using PL/SQL a mixture of a procedural language and SQL
  - called by invoking package name and handing parameters
  - add_booking (…….)

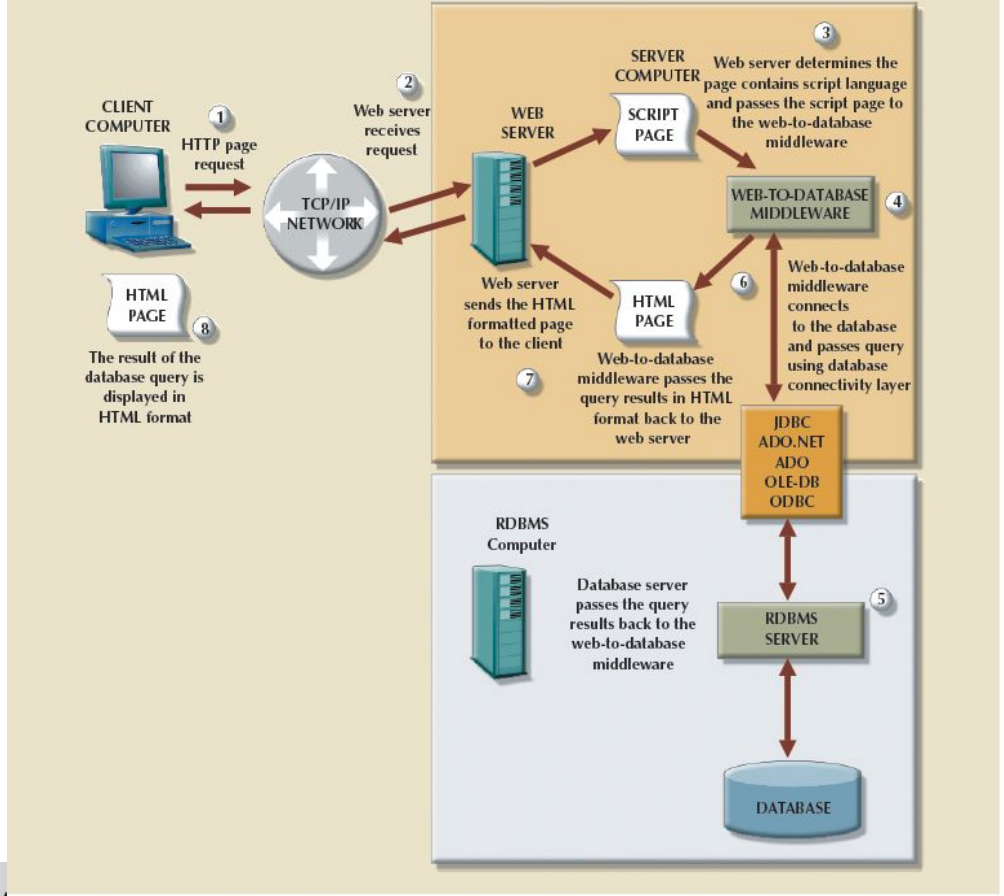
```
173     -- Procedure to add a new booking for a tour
174     PROCEDURE add_booking
175         (
176             arg_cust_no          IN book.cust_no%type,
177             arg_tour_no          IN book.tour_no%type,
178             arg_book_no_adults   IN book.book_no_adults%type,
179             arg_book_no_children IN book.book_no_children%type,
180             arg_booking_success OUT CHAR
181         )
182     AS
183
184         no_participants EXCEPTION;
185         already_booked  EXCEPTION;
186         tour_expired    EXCEPTION;
187         tour_no_space   EXCEPTION;
188
189         tourdatedepart  DATE;
190         tourmaxpartic   NUMBER;
191         totalchildren   NUMBER;
192         totaladults     NUMBER;
193         tourchildcost   NUMBER;
194         touradultcost   NUMBER;
195         tourbookcost    NUMBER;
196
197     BEGIN
198         arg_booking_success := '';
199
200         -- Check that some participants have been handed in for this booking
201         IF (arg_book_no_adults = 0) AND ( arg_book_no_children = 0) THEN
202             raise no_participants;
203         END IF;
204
205         -- Check customer, tour and booking validity
206
207         -- check_cust and tour are valid;
208         IF NOT valid_customer (arg_cust_no) THEN
209             raise invalid_customer;
```

Database connectivity - web technology

Q5. Which of the following are the roles of web to database middleware (multiple answers are possible):

a. Passes the query from the web server to the database middleware
b. Returns the query result from the database middleware back to the web server
c. Shows the HTML formatted page on a web browser
d. Compiles the PHP code

FIGURE 15.8 WEB-TO-DATABASE MIDDLEWARE



Coronel & Morris
Fig 15.8 Ed 13

Web Database Development

- Creating web pages which access data in a database. Many options available, including
  - ColdFusion Uses CFML - https://www.adobe.com/au/products/coldfusion-family.html or http://openbd.org/
  - PHP - http://php.net/
  - Oracle Application Express (Apex): https://apex.oracle.com/en/

TIOBE Index for Oct 2021 - Not Examinable

Oct 2021	Oct 2020	Change	Programming Language	Ratings	Change
1	3	↑	Python	11.27%	-0.00%
2	1	↓	C	11.16%	-5.79%
3	2	↓	Java	10.46%	-2.11%
4	4		C++	7.50%	+0.57%
5	5		C#	5.26%	+1.10%
6	6		Visual Basic	5.24%	+1.27%
7	7		JavaScript	2.19%	+0.05%
8	10	↑	SQL	2.17%	+0.61%
9	8	↓	PHP	2.10%	+0.01%
10	17	⇈	Assembly language	2.06%	+0.99%
11	19	⇈	Classic Visual Basic	1.83%	+1.06%
12	14	↑	Go	1.28%	+0.13%
13	15	↑	MATLAB	1.20%	+0.08%
14	9	⇊	R	1.20%	-0.79%
15	12	↓	Groovy	1.18%	-0.05%

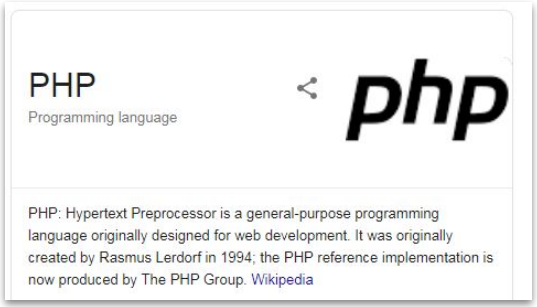
https://www.tiobe.com/tiobe-index/

PHP Basic

PHP Basic Case Study

- PHP language - server-side
  - –‘PHP-enabled web pages’ - https://www.php.net/manual/en/tutorial.php
  - –Commonly used in combination / part of frameworks (more later)

- PHP software needs to be alongside web server software
  - –e.g. besides Apache in LAMP stacks https://en.wikipedia.org/wiki/LAMP_(software_bundle);
  - –or PHP on IIS https://php.iis.net/
- **Further reading on PHP - “What can PHP do?”**
  - –https://www.php.net/manual/en/intro-whatcando.php



Example: Web Server and PHP

PHP Version 5.4.16

System	Linux [redacted] 3.10.0-862.el7.x86_64 #1 SMP Wed Mar 21 18:14:51 EDT 2018 x86_64
Build Date	Jan 23 2018 07:27:50
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d

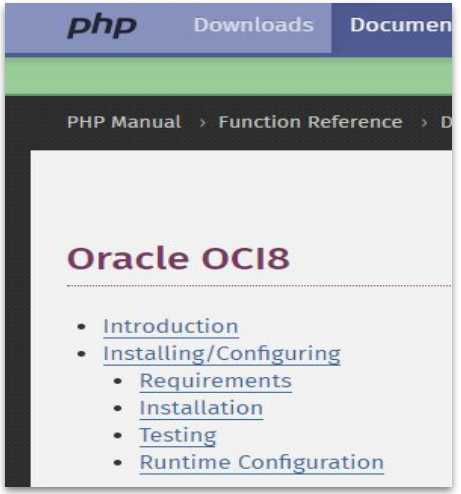
oci8

OCI8 Support	enabled
OCI8 DTrace Support	disabled
OCI8 Version	2.0.12
Revision	$Id: 020312b6429ebb9d6272ac9bc28f6dce529434b6 $
Oracle Run-time Client Library Version	12.1.0.2.0
Oracle Compile-time Instant Client Version	12.1

Directive	Local Value	Master Value
oci8.connection_class	*no value*	*no value*
oci8.default_prefetch	100	100
oci8.events	Off	Off
oci8.max_persistent	-1	-1
oci8.old_oci_close_semantics	Off	Off
oci8.persistent_timeout	-1	-1
oci8.ping_interval	60	60
oci8.privileged_connect	Off	Off
oci8.statement_cache_size	20	20

PHP Database Access

- PHP interacts with Oracle.
- Interaction via Oracle OCI 8 functions
  - Recommended reading: https://php.net/manual/en/book.oci8.php
  - Other RDBMS examples: PHP interacts with MySQL/MariaDB with **mysql_connect()** https://www.tutorialspoint.com/mariadb/mariadb_connection.htm
- Definition: **OCI8 is the PHP extension for connecting to Oracle Database**. OCI8 is open source and included with PHP. The name is derived from Oracle's C "call interface" API first introduced in version 8 of Oracle Database. OCI8 links with Oracle client libraries, such as Oracle Instant Client.



PHP Database Access Command - Connection

```
// Set up the Oracle connection string
  $dbInstance = "(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
    (HOST=ora-fit.ocio.monash.edu)(PORT=1521))
    (CONNECT_DATA=(SID=FITUGDB)))";

// Connect to the database - Open Oracle connection
  $conn = oci_connect($_POST["username"], $_POST["password"], $dbInstance);

  if (!$conn) {
      $e = oci_error();
      print "Error connecting to the database:<br>" ;
      print $e['message'] ;
      exit;
    }
```

PHP Database Access Command - Query Parsing

```
//SQL query statement
$query = "";

//Parse statement
$stmt = oci_parse($conn,$query);
if (!$stmt) {
  $e = oci_error($conn);
  print "Error on parse of statement:<br>" ;
  print $e['message'] ;
  exit;
}
```

PHP Database Access Command - Variable Mapping and Query Execution

```
// oci_define_by_name maps SQL Columns in a query to PHP variable names
// MUST be done before executing, Oracle names must be in UPPER case

      oci_define_by_name($stmt,"STUDID",$studid);
      ...

// Execute the STATEMENT
      $r = oci_execute($stmt);
      if (!$r) {
        $e = oci_error($stmt);
        print "Error execute of statement:<br>" ;
        print $e['message'] ;
        exit;
      }
```

PHP Database Access Command - Fetch Result and Close connection

```
// Fetch the results of the query
      while (oci_fetch($stmt)) {
        print("
        <tr>
          <td width=100>$studid</td>
          ...
        </tr>");
      }

// Free resources associated with Oracle statement
      oci_free_statement($stmt);

// Close the Oracle connection
      oci_close($conn);
```

The Web Page

Login to your Oracle Account

User Name:
Password:
Submit

Offering list UNIVERSITY database

Unit Code	Unit Name	Semester and Year	Chief Examiner
FIT1045	Algorithms and programming fundamentals in python	S1 2020	Gunar Dutch
FIT1045	Algorithms and programming fundamentals in python	S2 2019	Gunar Dutch
FIT1045	Algorithms and programming fundamentals in python	S1 2019	Gunar Dutch
FIT1050	Web fundamentals	S1 2019	Sandro Wethered
FIT1050	Web fundamentals	S2 2019	Sandro Wethered
FIT1050	Web fundamentals	S1 2020	Sandro Wethered
FIT2094	Databases	S1 2020	Lizabeth Stubbings
FIT2094	Databases	S2 2019	Lizabeth Stubbings
FIT3157	Advanced web design	S1 2020	Trixy Warner
FIT3157	Advanced web design	S2 2019	Trixy Warner
FIT3176	Advanced database design	S1 2020	Windham Ellard
FIT3176	Advanced database design	S2 2019	Windham Ellard
FIT5145	Introduction to data science	S2 2019	Windham Ellard
FIT5145	Introduction to data science	S1 2020	Windham Ellard
FIT5196	Data wrangling	S2 2019	Windham Ellard
FIT5196	Data wrangling	S1 2020	Windham Ellard
FIT9132	Introduction to databases	S2 2019	Xena Epine
FIT9132	Introduction to databases	S1 2019	Xena Epine
FIT9132	Introduction to databases	S1 2020	Xena Epine
FIT9136	Algorithms and programming foundations in Python	S1 2020	Tammi Soane
FIT9136	Algorithms and programming foundations in Python	S1 2019	Tammi Soane
FIT9136	Algorithms and programming foundations in Python	S2 2019	Tammi Soane
FIT9137	Introduction to computer architecture and networks	S1 2020	Kennie Pickin
FIT9137	Introduction to computer architecture and networks	S2 2019	Kennie Pickin

Rows found:24

Practical considerations and security

Use of Frameworks

- Earlier we discussed the fact that PHP is used within many frameworks
  - So what are frameworks?
- "A web framework (WF)... is a software framework that is designed to support the development of web applications …
  - "[they] provide a standard way to build and deploy web applications on the World Wide Web… automate the overhead associated with common activities performed in web development. …
  - "[e.g.] provide libraries for database access"

  https://en.wikipedia.org/wiki/Web_framework
- Trends in 2021 - see e.g.
  - https://hackr.io/blog/top-10-web-development-frameworks-in-2020

Frameworks, Oracle Support, ORM

- Many frameworks support Oracle connectivity.
- Examples:
  - Django https://docs.djangoproject.com/en/2.2/ref/databases/
  - Node.js https://www.oracle.com/au/database/technologies/appdev/nodejs.html
  - CakePHP https://github.com/CakeDC/cakephp-oracle-driver
  - Symfony https://symfony.com/doc/current/doctrine.html
- Object-Relational Mapping (ORM) helps make it easy to write code ...
  - A short definition: “Object-Relational Mapping is a technique that lets you query and manipulate... data from a database using an object-oriented paradigm.” Reference: https://blog.yellowant.com/orm-rethinking-data-as-objects-8ddaa43b1410
  - Shorter example: CakePHP’s ORM maps a DB row to an object in your programming language of choice (e.g. **$article** in CakePHP)...
    - so you can use the object directly to access its attributes e.g. **$article->title**

SQL Injection Demo

https://www.w3schools.com/sql/sql_injection.asp

Security Considerations

- Databases, especially when they are user-facing (web apps etc), are at risk of attacks over the web...
  - **OWASP's Top 10 list since 2010 to 2017 -- #1 is "Injection"**
  - Read https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project
- SQL injection is very common! Definition: quoted verbatim (OWASP)
  - "A SQL injection attack consists of insertion or "injection" of a SQL query via the input data from the client to the application. A successful SQL injection exploit can read sensitive data from the database, modify database data (Insert/Update/Delete), execute administration operations on the database (such as shutdown the DBMS), recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system. SQL injection attacks are a type of injection attack, in which SQL commands are injected into data-plane input in order to effect the execution of predefined SQL commands." https://www.owasp.org/index.php/SQL_Injection
  - (OWASP: Open Web Application Security Project)

Q6. To prevent SQL injection (multiple answers are possible):

a. use views
b. create stored procedures
c. sanitise and check the user input
d. grant the same privilege to all users

Security Considerations

- Lessons:
  - Sanitise and check your input!
  - Configure your database to minimise the damage
    - restricted user - least privileges
    - using views (Workshop 10)
  - Follow security best practices
    - e.g. OWASP https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.md
    - e.g. for Oracle -
      - Oracle Blog https://blogs.oracle.com/sql/what-is-sql-injection-and-how-to-stop-it
      - 67-page whitepaper https://www.oracle.com/assets/how-to-write-injection-proof-plsql-1-129572.pdf

Cloud Computing


What is
Cloud
Computing?

Cloud Computing

"A computing model for enabling **ubiquitous**, convenient, **on-demand** network **access to a shared pool of configurable computer resources** (e.g., networks, servers, storage, applications and services) that can be **rapidly provisioned** and released with **minimal management effort** or service provider interaction."

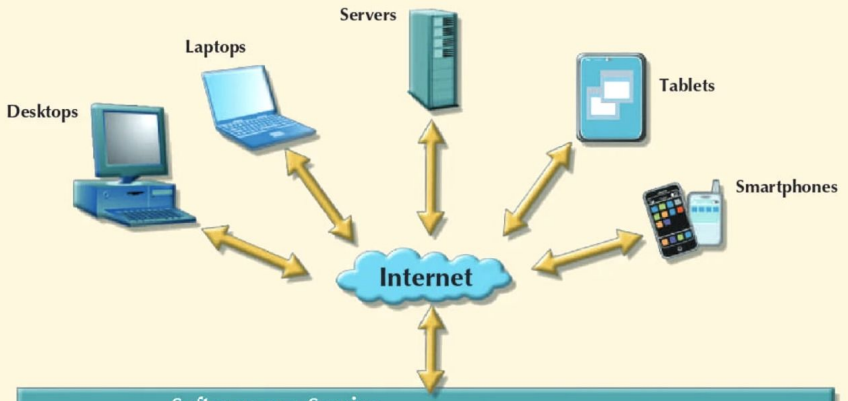
Cloud implementation types:

- Public Cloud
- Private Cloud
- Community Cloud

Characteristic of Cloud Services

- Ubiquitous access via Internet technologies
- Shared infrastructure.
- Lower startup costs and variable pricing
- Flexible and scalable services
- Dynamic provisioning
  - e.g. use the web management dashboard to add and remove services on demand
- Service orientation
- Managed operations
  - minimizes the need for extensive and expensive in-house IT staff

FIGURE 15.23 TYPES OF CLOUD SERVICES



Coronel & Morris
Fig 15.23 Ed 13

TABLE 15.4

ADVANTAGES AND DISADVANTAGES OF CLOUD COMPUTING

ADVANTAGE	DISADVANTAGE
Low initial cost of entry. Cloud computing has lower costs of entry when compared with the alternative of building in house.	*Issues of security, privacy, and compliance*. Trusting sensitive company data to external entities is difficult for most data-cautious organizations.
Scalability/elasticity. It is easy to add and remove resources on demand.	*Hidden costs of implementation and operation*. It is hard to estimate bandwidth and data migration costs.
Support for mobile computing. Cloud computing providers support multiple types of mobile computing devices.	*Data migration is a difficult and lengthy process*. Migrating large amounts of data to and from the cloud infrastructure can be difficult and time-consuming.
Ubiquitous access. Consumers can access the cloud resources from anywhere at any time, as long as they have Internet access.	*Complex licensing schemes*. Organizations that implement cloud services are faced with complex licensing schemes and complicated service-level agreements.
High reliability and performance. Cloud providers build solid infrastructures that otherwise are difficult for the average organization to leverage.	*Loss of ownership and control*. Companies that use cloud services are no longer in complete control of their data. What is the responsibility of the cloud provider if data are breached? Can the vendor use your data without your consent?
Fast provisioning. Resources can be provisioned on demand in a matter of minutes with minimal effort.	*Organization culture*. End users tend to be resistant to change. Do the savings justify being dependent on a single provider? Will the cloud provider be around in 10 years?
Managed infrastructure. Most cloud implementations are managed by dedicated internal or external staff. This allows the organization's IT staff to focus on other areas.	*Difficult integration with internal IT system*. Configuring the cloud services to integrate transparently with internal authentication and other internal services could be a daunting task.

Coronel & Morris
Table 15.4 Ed 13