MONASH University

MONASH
INFORMATION
TECHNOLOGY

Week 11 -
Database Web Interfaces

FIT2094 - FIT3171 Databases
Clayton Campus S1 2019.

GROUP OF EIGHT AUSTRALIA

Where Are We

- Through this unit we have looked at
  - The fundamental principles on which relational databases are built
  - How we design a database
  - How we create objects in a relational database and manipulate 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
  - GRANT used to control access to this accounts objects (like `SHAREDSAMPLES` account in Tute 8)

Overview

- **Hour 1**
  - –Database connectivity
    - •incl. middleware, ODBC, JDBC etc
  - –Database web connectivity
    - •development
    - •TIOBE Index of common languages

… then COFFEE BREAK!

- **Hour 2**
  - –PHP Basic case study
    - •OCI 8 and general techniques
  - –Practical considerations and security
    - •Frameworks and ORM
    - •SQLi

Q1. The interface between an application program and the database, is known as

a. SQL
b. Database Middleware
c. The Data Layer
d. A Client Side Extension
e. Data Access Objects

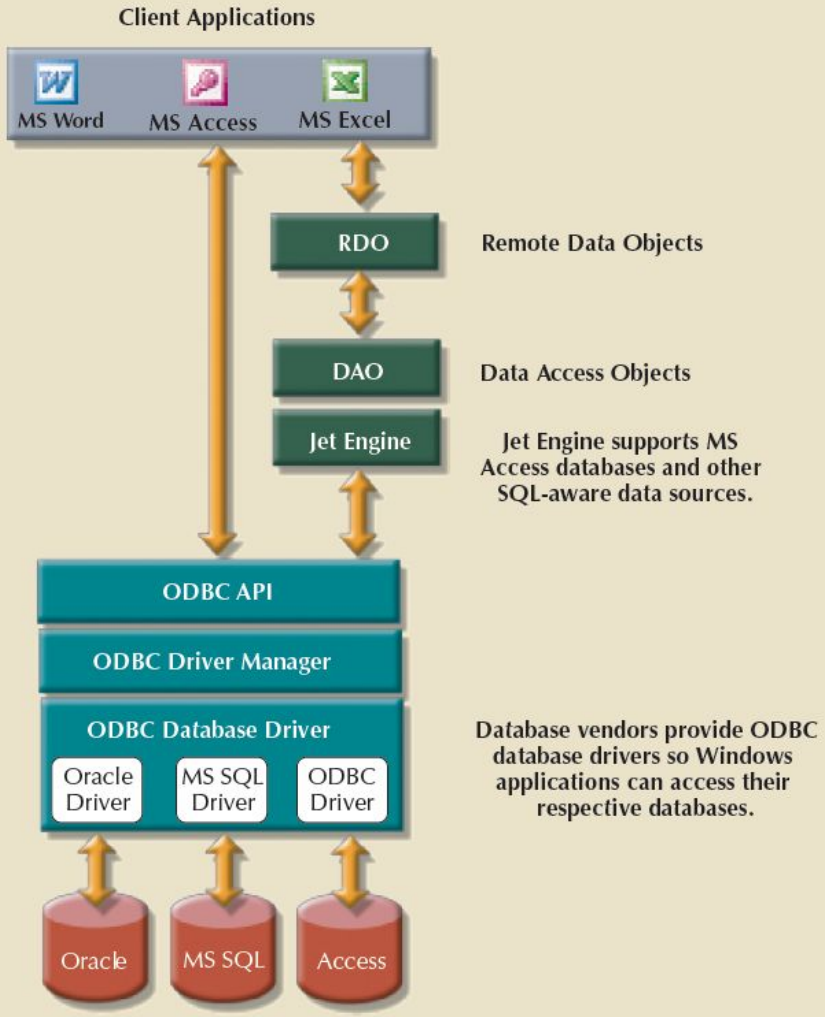
Database Connectivity

Img src: @jankolar at Unsplash

Database Connectivity

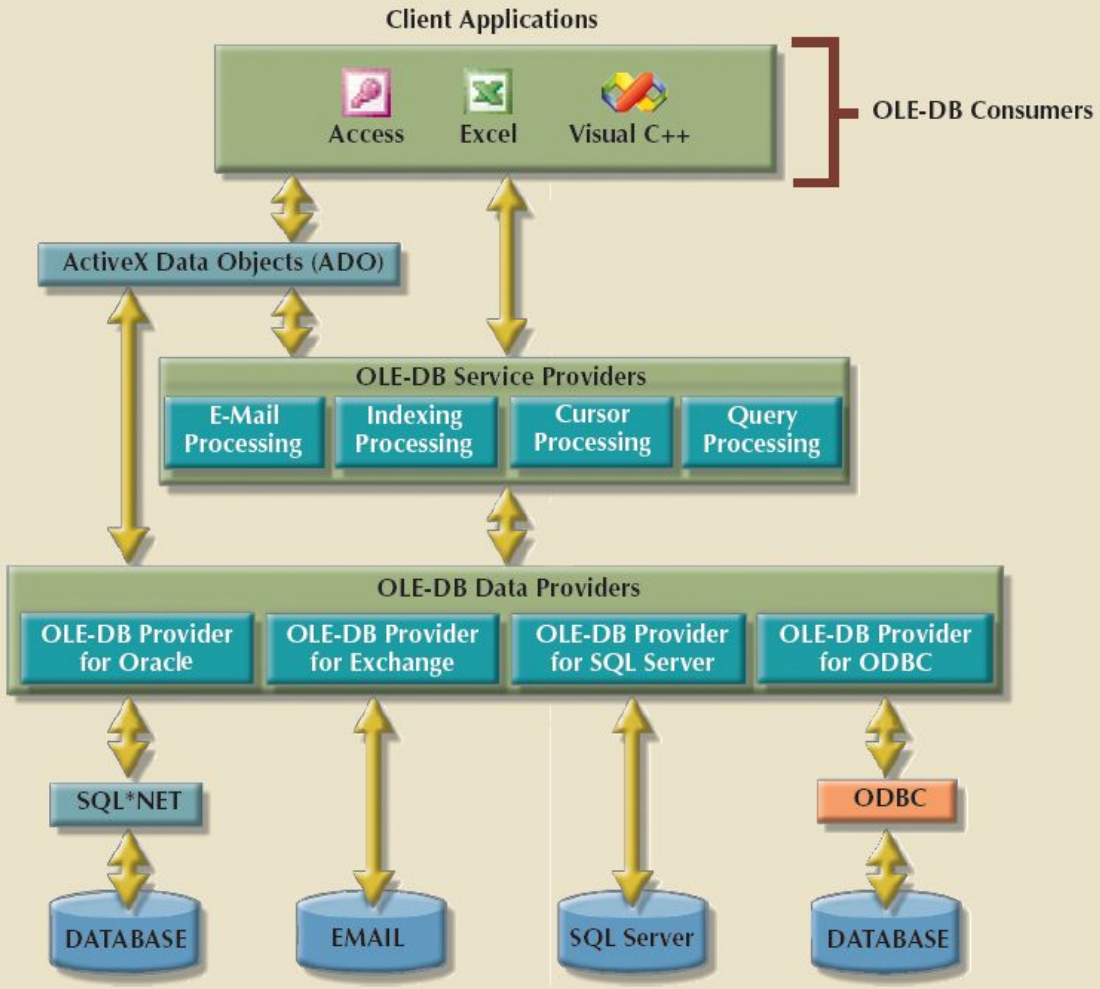
- The DATA LAYER – your data management application (DBMS)
- The DATABASE MIDDLEWARE – manages connectivity and data transformation issues. Many options available such as:
  - Native SQL Connectivity
    - Vendor provided eg. Oracle SQL*Net
  - Microsoft ODBC, DAO, RDO; OLE-DB and ADO.NET
  - Java Database 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



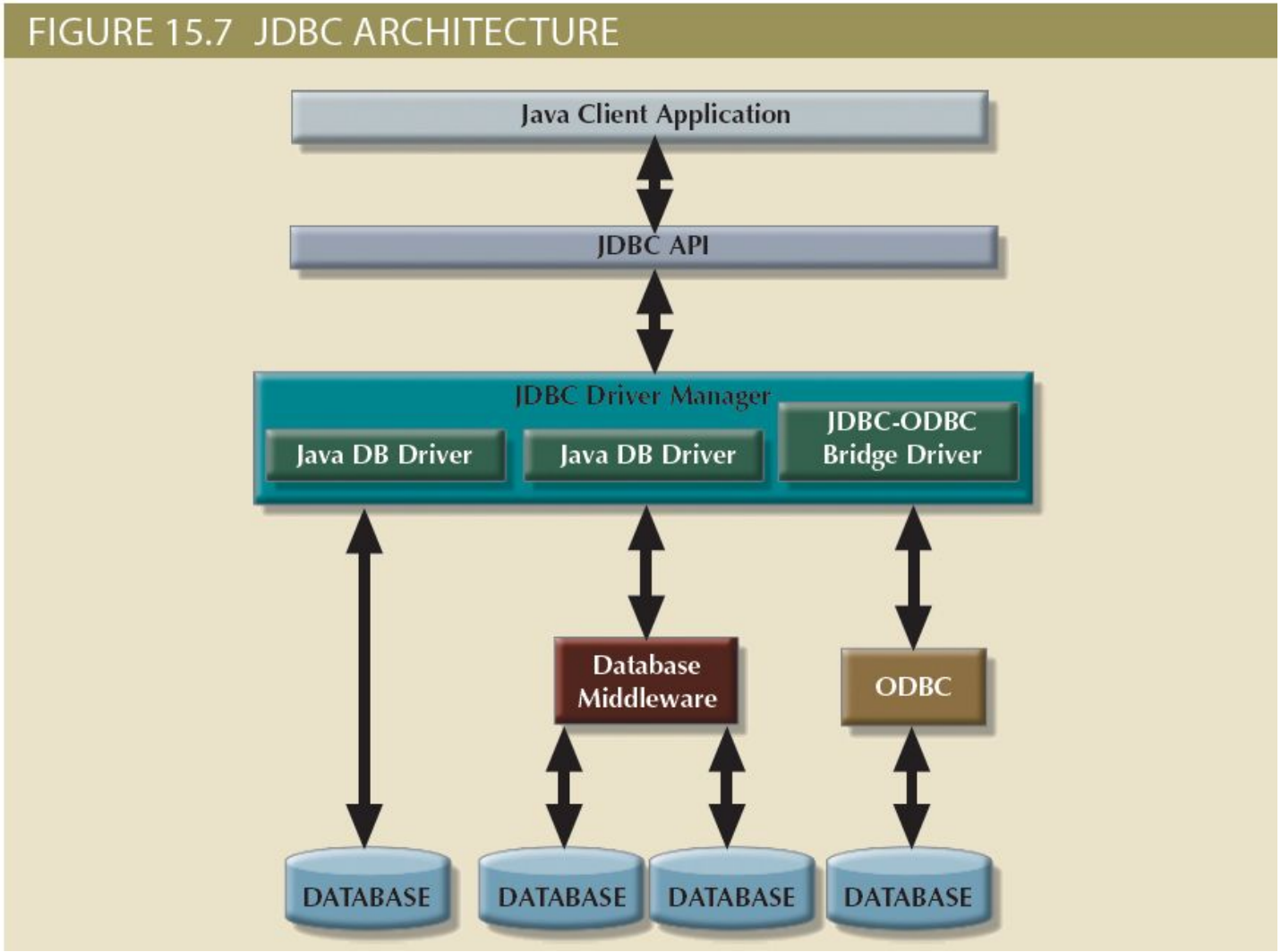
Coronel & Morris
Fig 15.2 Ed 13

FIGURE 15.5 OLE-DB ARCHITECTURE



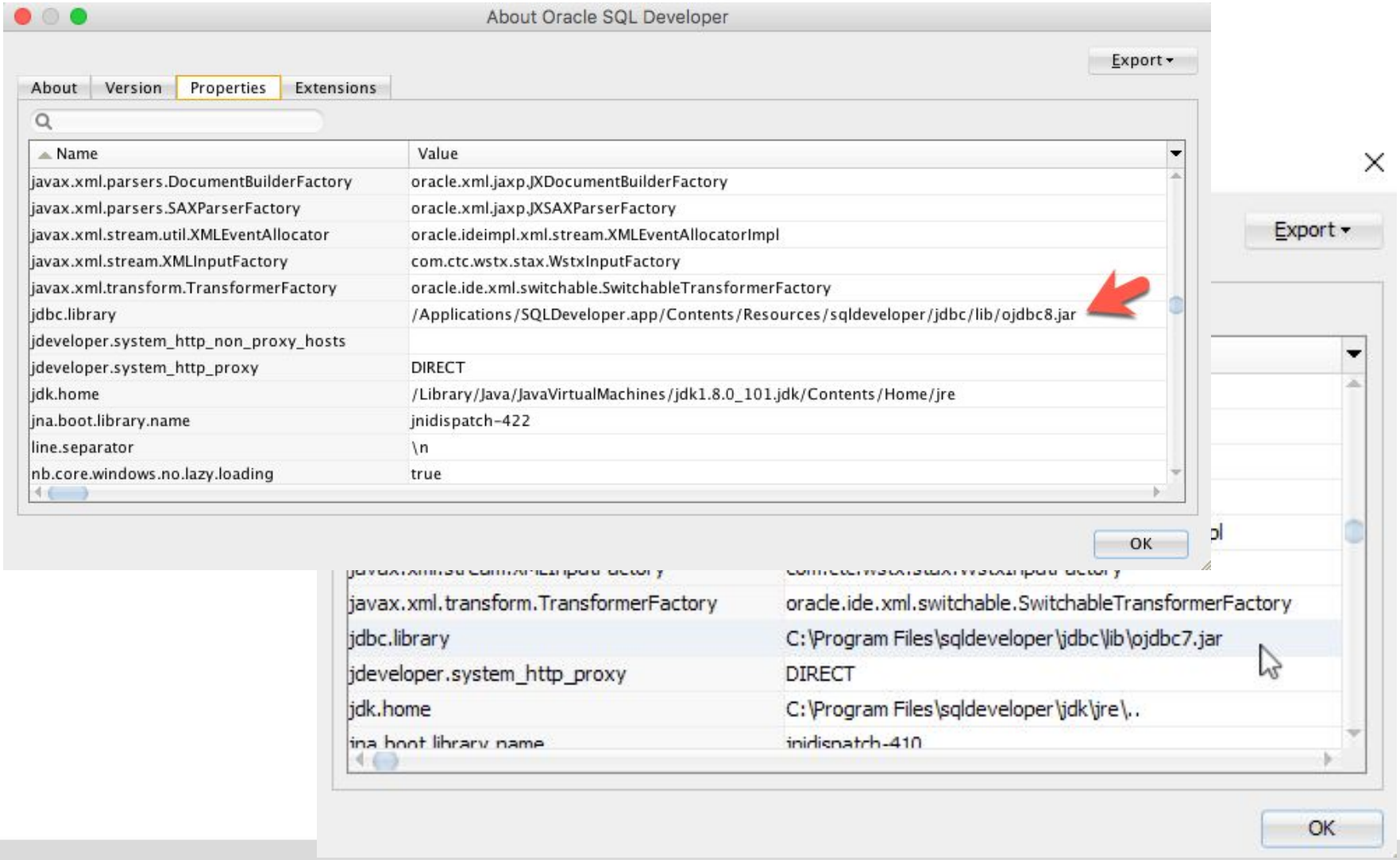
Coronel & Morris
Fig 15.5 Ed 13

FIGURE 15.7 JDBC ARCHITECTURE



Coronel & Morris
Fig 15.7 Ed 13

[Lindsay] Anecdote: 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://goo.gl/8l8R

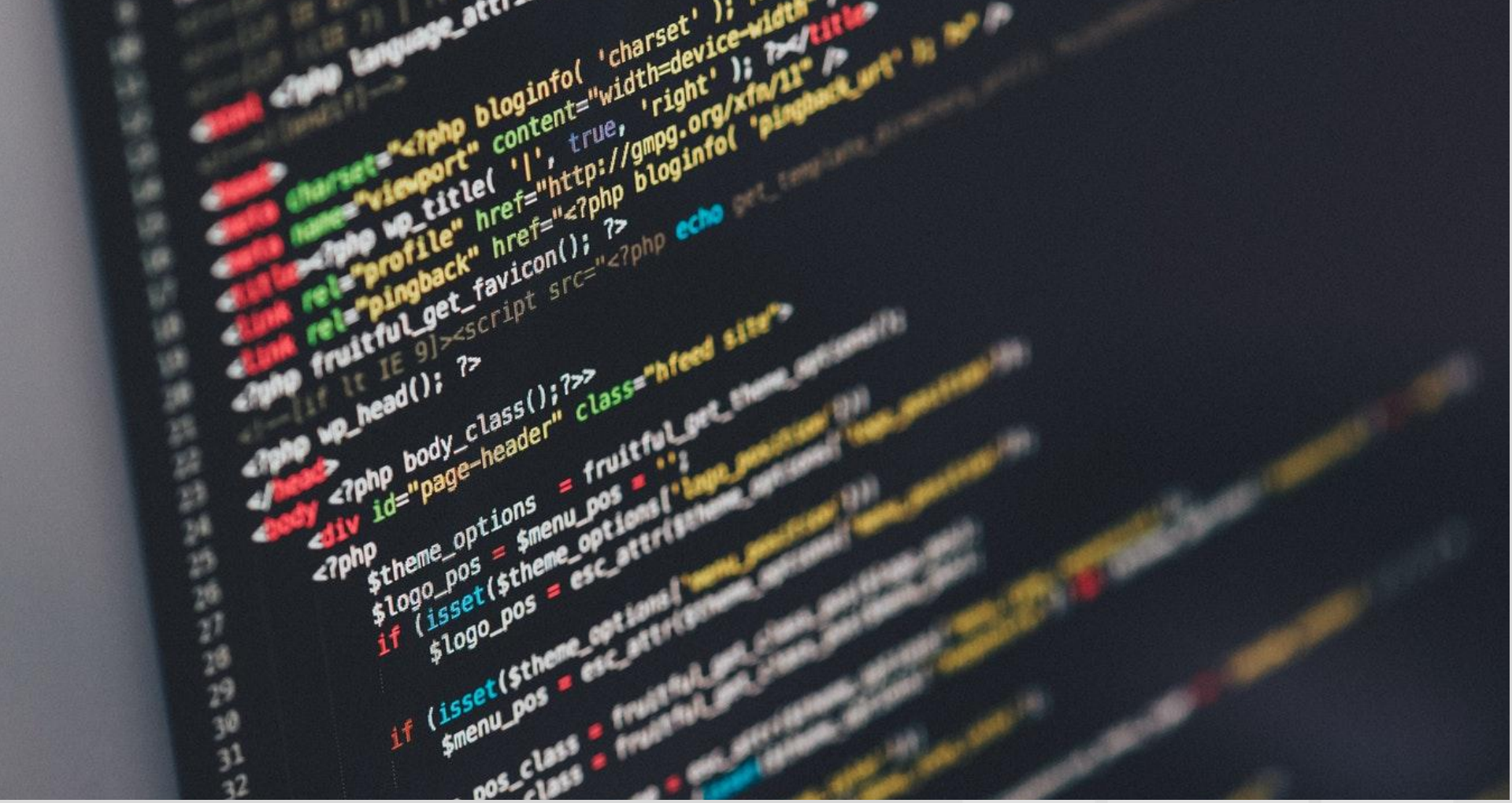
Placing application logic in the backend

- 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 (…….)
- Covered in Advanced Database FIT3176
- http://www.monash.edu/pubs/2019handbooks/units/FIT3176.html

```
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;
```

PL/SQL Triggers and DB Usage - Recap

- Speaking of PL/SQL: Recall Week 9 on PL/SQL Triggers.
- As your users (and their applications) access your DB, it is important to make sure triggers are coded properly
  - –for auditing
  - –to comply with business rules (e.g. cascading updates) etc
- **Friendly reminder: BOTH FIT2094 and FIT3171 have to study PL/SQL Triggers well.**
- **FIT3171 WILL be examined more rigorously on Oracle Triggers due to ULO #7…**
  - **–This means more quality/quantity of Q's…**
  - **–ULO "7. develop … with a database backend;"**



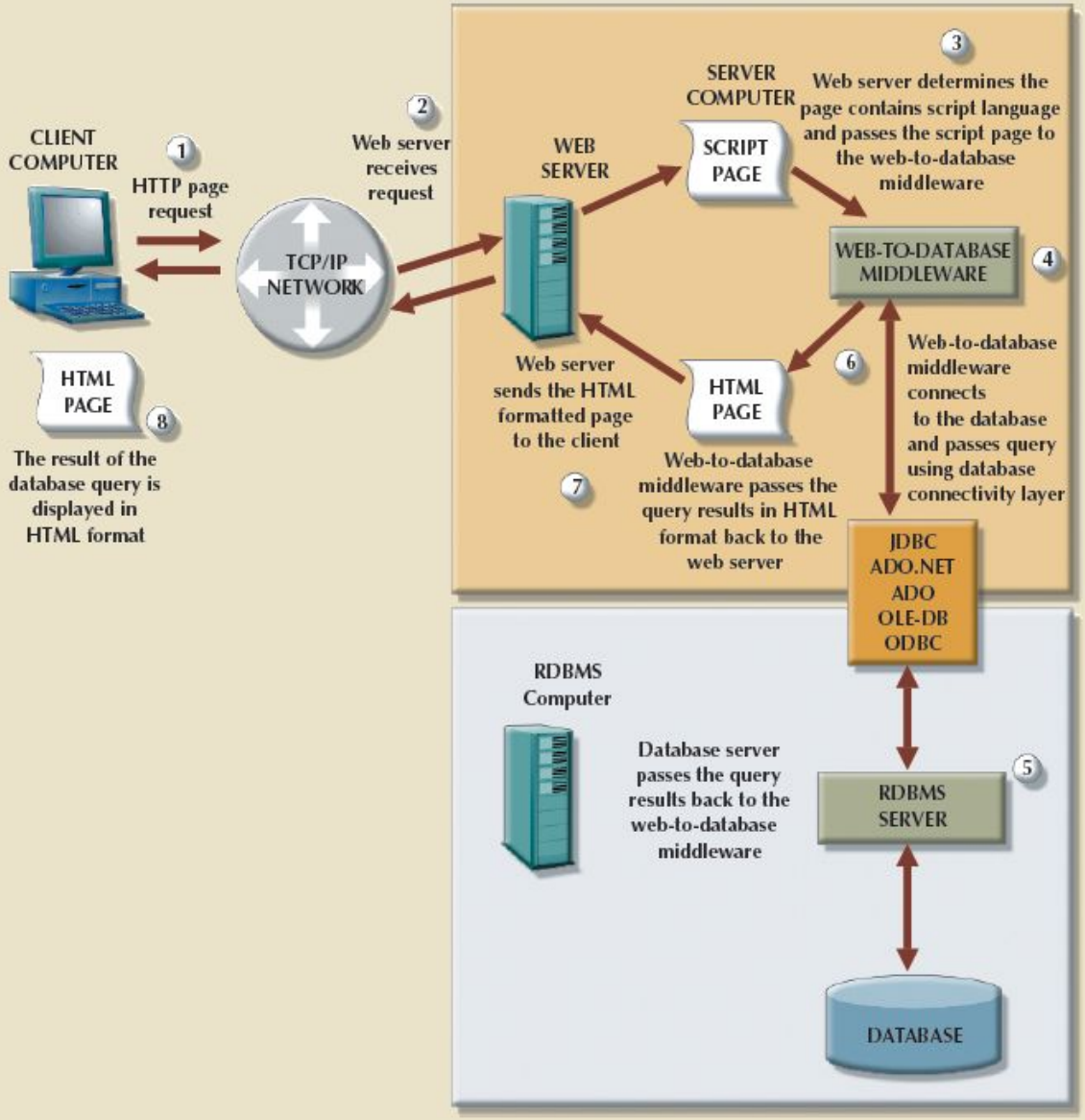
Database Internet Connectivity

Img src: @ilyapavlov at Unsplash

Q2. A server-side extension is

a. part of web server which allows it to be used across many hosts
b. is necessary to access a web server from a mobile device
c. a program that interacts directly with the web server to handle specific types of requests
d. interacts directly with a client-side extension
e. a vendor specific approach to accessing a database across the internet

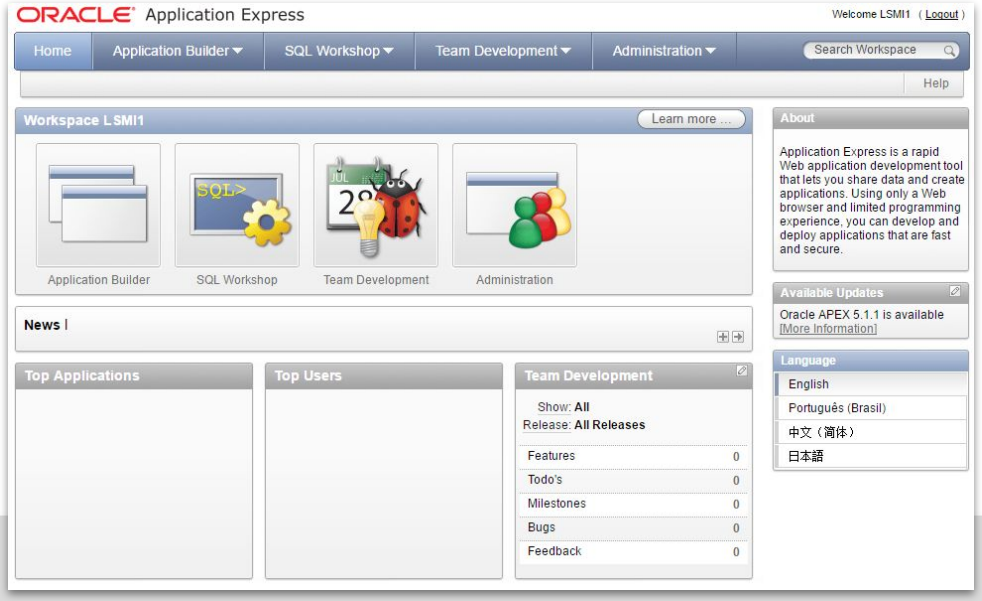
FIGURE 15.8 WEB-TO-DATABASE MIDDLEWARE



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://goo.gl/7FnYgi or http://openbd.org/
  - PHP - http://php.net/
  - Oracle Application Express (Apex)



[Lindsay's anecdote]
TIOBE Index for October 2018

Oct 2018	Oct 2017	Change	Programming Language	Ratings	Change
1	1		Java	17.801%	+5.37%
2	2		C	15.376%	+7.00%
3	3		C++	7.593%	+2.59%
4	5	^	Python	7.156%	+3.35%
5	8	^	Visual Basic .NET	5.884%	+3.15%
6	4	v	C#	3.485%	-0.37%
7	7		PHP	2.794%	+0.00%
8	6	v	JavaScript	2.280%	-0.73%
9	-	^^	SQL	2.038%	+2.04%
10	16	^^	Swift	1.500%	-0.17%
11	13	^	MATLAB	1.317%	-0.56%
12	20	^^	Go	1.253%	-0.10%
13	9	vv	Assembly language	1.245%	-1.13%
14	15	^	R	1.214%	-0.47%
15	17	^	Objective-C	1.202%	-0.31%

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

TIOBE Index for 2019

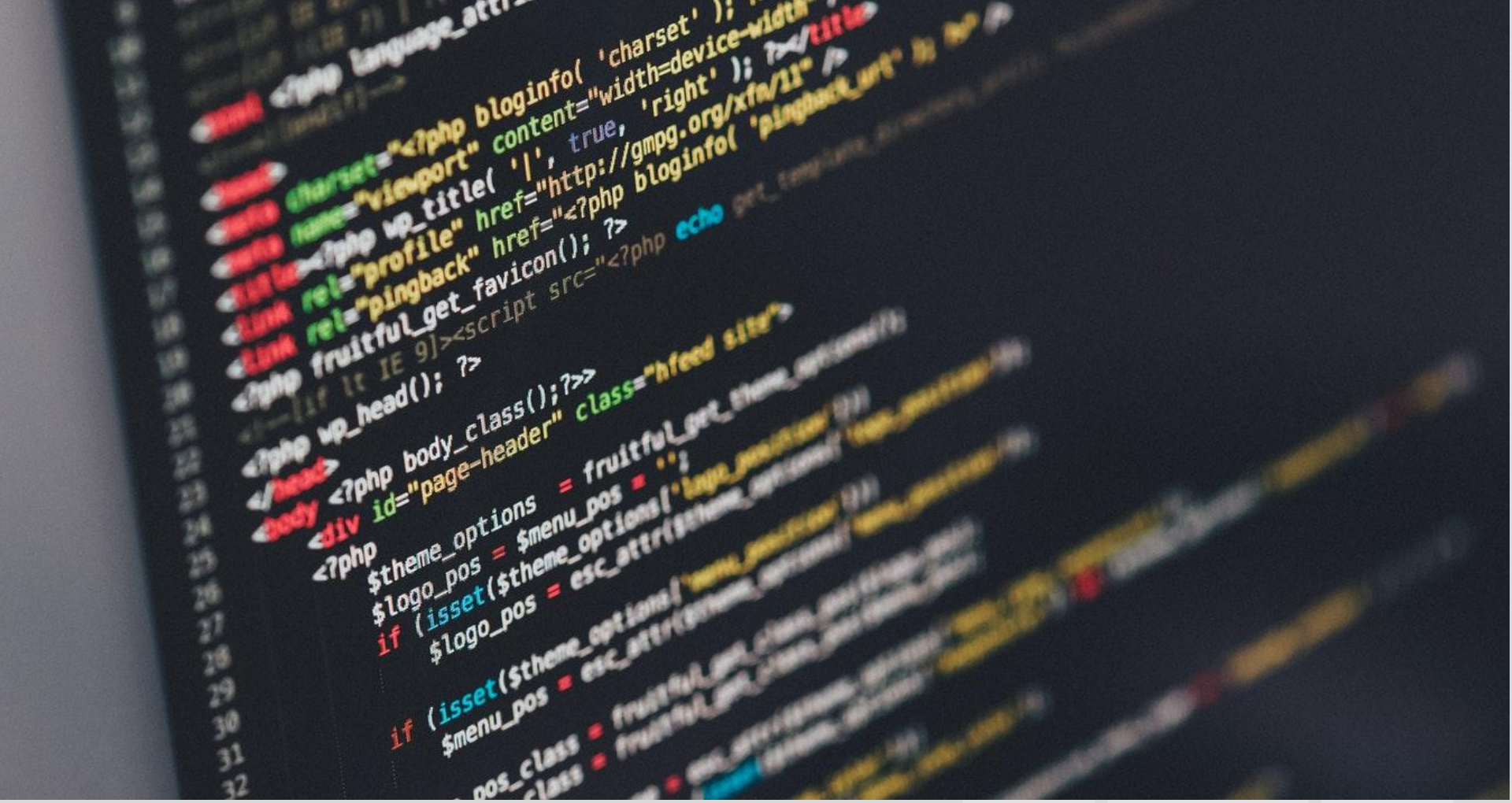
The index can be used to check whether your programming skills are still up to date or to make a strategic decision about what programming langu should be adopted when starting to build a new software system. The definition of the TIOBE index can be found here.

May 2019	May 2018	Change	Programming Language	Ratings	Change
1	1		Java	16.005%	-0.38%
2	2		C	14.243%	+0.24%
3	3		C++	8.095%	+0.43%
4	4		Python	7.830%	+2.64%
5	6	^	Visual Basic .NET	5.193%	+1.07%
6	5	v	C#	3.984%	-0.42%
7	8	^	JavaScript	2.690%	-0.23%
8	9	^	SQL	2.555%	+0.57%
9	7	v	PHP	2.489%	-0.83%
10	13	^	Assembly language	1.816%	+0.82%

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

Coffee break - see you in 10 minutes.

Img src: Jazmin Quaynor @jazminantoinette at Unsplash



Database Internet Connectivity: PHP Case Study

Img src: @ilyapavlov at Unsplash

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

PHP Basic Case Study

- Quick synopsis
  - –When a PHP page is accessed, PHP interpreter living in the server produces output, which is served to the user (commonly interpreted in the user's browser as HTML). Users don't see the raw PHP code.
- "... when PHP is installed, the web server is configured to expect certain file extensions to contain PHP language statements. ... **When the web server gets a request for a file with the designated extension**, it **sends the HTML statements as is**, but **PHP statements are processed by the PHP software before they're sent to the requester**… When PHP language statements are processed, **only the output, or anything printed to the screen is sent by the web server to the web browser**."
  - –Source: Suehring & Valade. Read the full article: https://www.dummies.com/programming/php/how-php-works/

Q3. PHP is

a. a piece of software which lives on the server
b. an RDBMS library itself
c. a programming/scripting language
d. owned by Oracle
e. all of (a, b, c)
f. both (a, c)
g. both (c, d)

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."
  - https://www.oracle.com/technetwork/articles/technote-php-instant-084410.html

Note on PHP/OCI8 and the exam

- For this unit we are not expecting you to become PHP experts.

- **You are not expected to know how to code in PHP (writing code, correct syntax, etc) for the exam**
- **...but you are expected to know some basic theory of how to develop an interface for databases.**
  - (including, but not limited to, the very basics on how PHP works with OCI, good practices, etc).

PHP Database Access - Set up credentials

- Login details are coded in PHP in a separate file.
  - allows login code reuse in a central area (folder). Developers must make sure nobody else can access these secure credentials.
- **Set up login** details:

```
$dbUserName = "username";    //database user
$dbPassword = "whatever";    //database password
$dbInstance="connection string"; // see below
```

- Oracle Connection String ($dbInstance) is a special string specifying certain connection details. HOST and SID are the server address and SID respectively.

```
$dbInstance = "(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
    (HOST=oracle.server.example.com)(PORT=1521))
    (CONNECT_DATA=(SID=databaseSID)))";
```

PHP Database Access - oci_connect

- oci_connect is used to **open a database connection**,

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

- This sample PHP code reports an error to browser and exits if connection is not created successfully.
  - –via checking the connection 'identifier' in the IF statement
  - –https://www.php.net/manual/en/function.oci-connect.php

PHP Database Access - simple interface to display tables

Steps

1. Create HTML table header (output to be placed in a table) for browser to render.
   a. this can be plain HTML; e.g. →

```
<!-- create HTML table header to display output -->
<table border =1 width=600>
<tr>
 <th width=100><b>Student ID</b></td>
 <th width=200><b>Name</b></td>
 <th width=100><b>Birth Date</b></td>
 <th width=200><b>Email</b></td>
</tr>
```

2. **Build Query String** ($query)
   a. in PHP, this string variable is a SQL select statement
   b. simple example here →

```
<?php
//SQL query statement
$query = "SELECT a,b,c,d FROM table ORDER BY c";
```

3. **Parse statement** (SQL select in $query string)

```
$stmt = oci_parse($conn,$query);
```

PHP Database Access - simple interface to display tables

4. **Map** the Oracle SQL Columns → PHP variables
   a. "Associates a PHP variable with a column for query fetches using oci_fetch()... The oci_define_by_name() call must occur before executing oci_execute()." https://www.php.net/manual/en/function.oci-define-by-name.php
   b. e.g. `oci_define_by_name($stmt,"SNAME",$stuname);` maps **SNAME** in Oracle to var **$stuname** in PHP

5. **Execute** the statement
   a. `$r = oci_execute($stmt);`

6. **Fetch the results** of the Query
   a. in a loop - use **oci_fetch** to iterate through each row https://www.php.net/manual/en/function.oci-fetch.php
   b. Simple example →

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

PHP Database Access - Discussion

- Make sure HTML produced by the PHP code is syntactically correct
  - e.g. 'bold' tag not output properly

```
<?php
 print("<b>$variable</b");
>?
```

- Make sure the HTML file outside of PHP code is correct
  - e.g. closing tags for </html>; check inline Javascript, CSS
- All the oci_ statements covered in PHP's manual
  - https://www.php.net/manual/en/ref.oci8.php

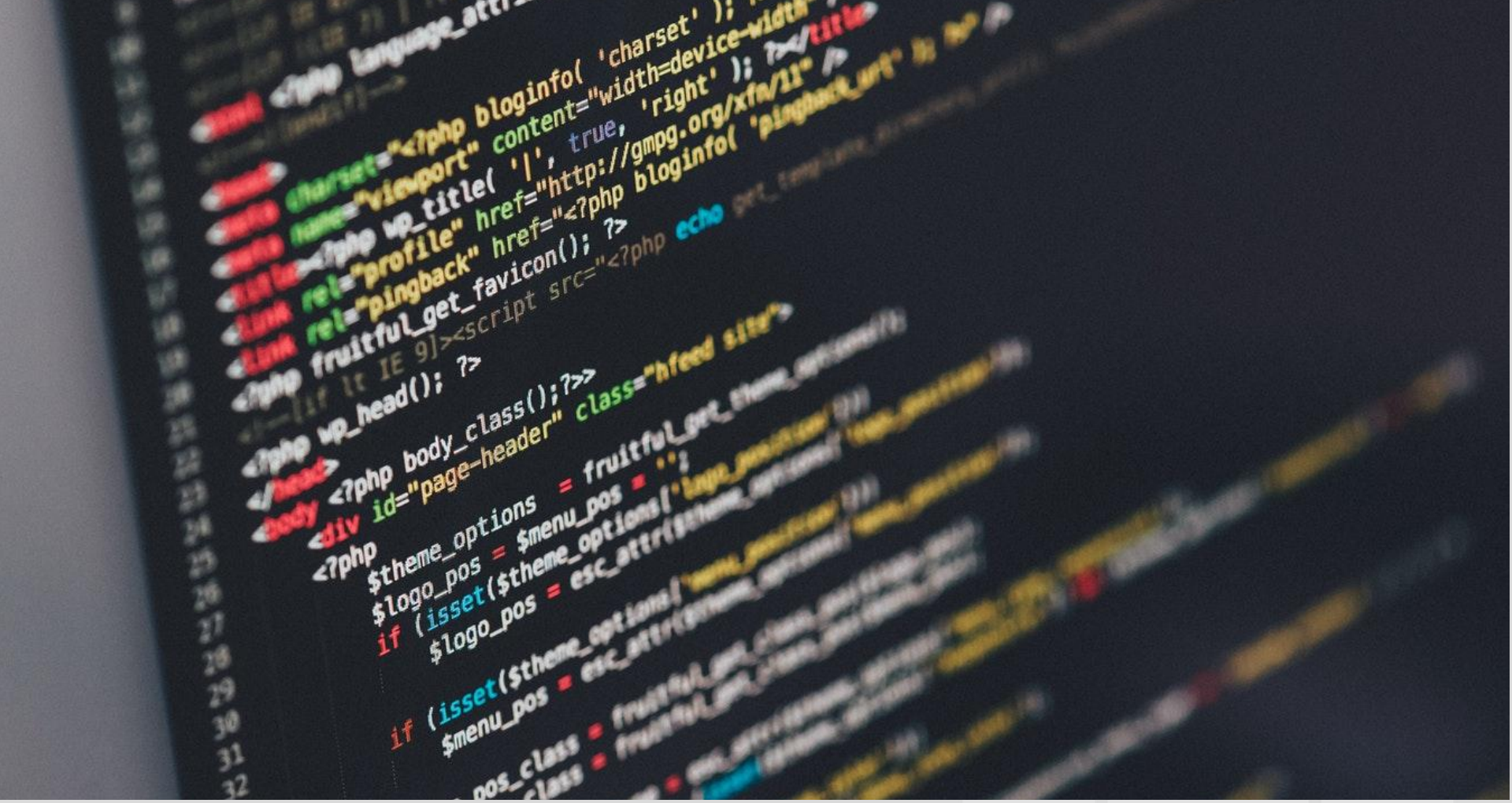
- The Tute in Week 12 contains sample code (from start to finish) on a simple interface.
- **Remember: coding and setting up a PHP server / database server is BEYOND the scope of the unit.**
  - you ARE NOT expected e.g. to code the web app in the exam!

PHP Database Access - sample output

Student list UNIVERSITY database

Student ID	Name	Birth Date	Email
11111111	Mary Smith	01-Jan-1995	[redacted]@monash.edu
11111112	Matthew Long	01-Feb-1997	[redacted]@monash.edu
11111113	Andy Lee	01-Mar-1995	[redacted]monash.edu
11111114	Rani Dewa	01-Apr-1996	[redacted]@monash.edu
11111115	David Dumbledore	02-Jan-1996	[redacted]@monash.edu
11111116	John Chung	03-Dec-1996	[redacted]@monash.edu
11111117	Jake Ryan	01-Jan-1990	[redacted]monash.edu
11111118	Theo Gupta	12-Jul-1992	[redacted]@monash.edu
11111119	Samuel Nguyen	15-Sep-1996	[redacted]n@monash.edu
11111120	James Dowe	01-Jan-1996	[redacted]@monash.edu
11111121	Mary Chan	01-Jan-1991	[redacted]@monash.edu
11111122	Andrew Short	01-Feb-1990	[redacted]@monash.edu
11111123	Tay Lee	01-Mar-1988	[redacted]monash.edu
11111124	Dani Solo	01-Apr-1991	[redacted]@monash.edu
11111125	David Smith	02-Jan-1990	[redacted]@monash.edu
11111126	John Tse	03-Dec-1988	[redacted]@monash.edu
11111127	Jake Brown	01-Jan-1990	[redacted]monash.edu
11111128	Gary Gupta	12-Jul-1992	[redacted]@monash.edu
11111129	Ruth Nguyen	15-Sep-1991	[redacted]n@monash.edu
11111130	Blake White	01-Jan-1992	[redacted]@monash.edu

Rows found:20



Database Internet Connectivity: Practical Considerations and Security

Img src: @ilyapavlov at Unsplash

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 2019 - see e.g.
  - https://hackernoon.com/top-10-python-web-frameworks-to-learn-in-2018-b2ebab969d1a
  - Image source: Goel (2019)

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 - User Agent

```
GET / HTTP/1.1
host: www.example.com
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (compatible; MSIE 11.0; Windows NT 6.1; Win64; x64;
Trident/5.0)'+(select*from(select(sleep(20)))a)+'
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8,fr;q=0.6
```

```
<?php
$link = new mysqli('localhost', 'insecure', '1ns3cur3p4ssw0rd', 'analytics');
$query = sprintf("INSERT INTO visits (ua, dt) VALUES ('%s', '%s')", $_SERVER["HTTP_USER_AGENT"]
                                                                  , date("Y-m-d h:i:s"));
$link->query($query);
?>
<html><head></head><body><b>Thanks for visiting</b></body></html>
```


Src: https://securityonline.info/sql-injection-user-agent-injection-attack/

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)

Security Considerations

- Examples -
  - simple ones illustrated in https://www.w3schools.com/sql/sql_injection.asp

- Lessons:
  - Sanitise and check your input!
  - Configure your database to minimise the damage
    - restricted user - least privileges
    - using views (Lecture 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

Q4. Given the following SQL statement in the back-end:

```
SELECT name, company, phone FROM vendors
WHERE  name = '$variable';
```

What can go wrong if SQL is injected via $variable e.g. on a web form?

a. tables can be DROPped
b. ALTERations can be done to tables
c. vendor names can be UPDATED
d. potentially sensitive data e.g. logins in a secret table can be UNIONed
e. All of the above
f. None of the above