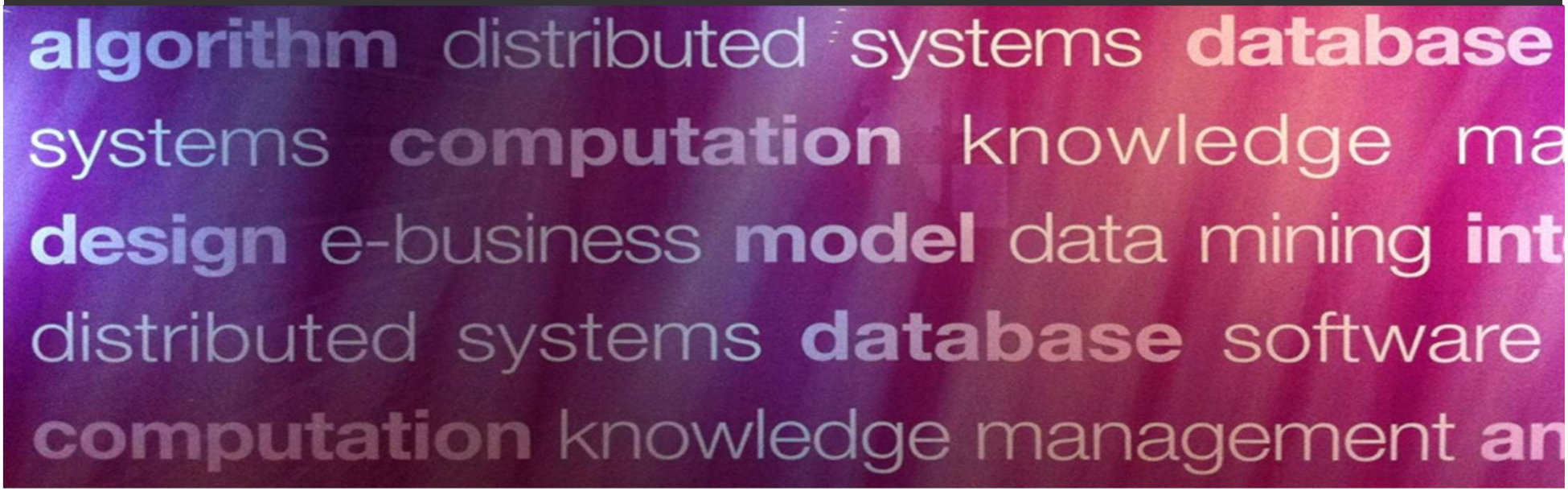
MONASH University

Information Technology

FIT2094 Databases

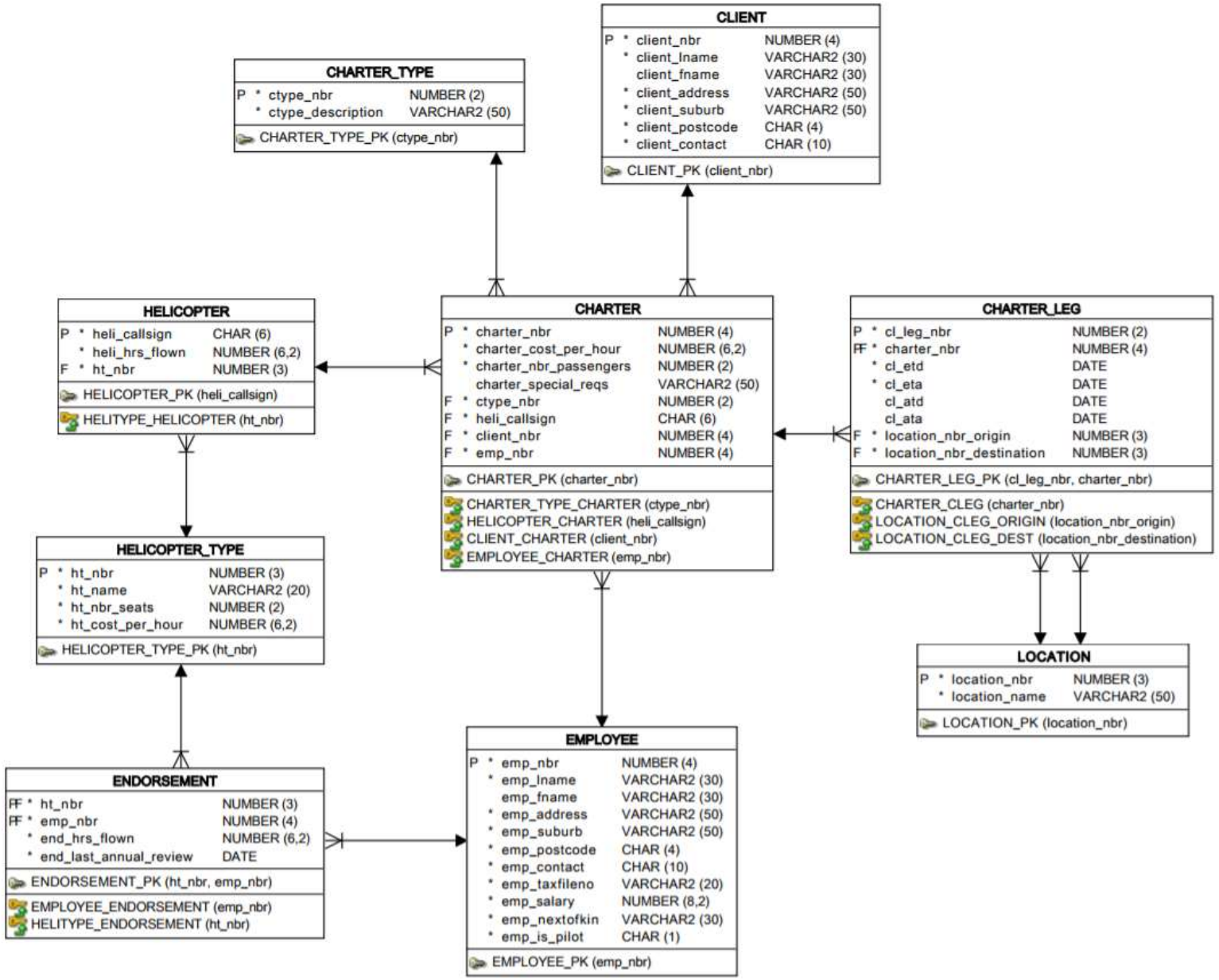
Week 11 – Database Web Interfaces



Week 12 Unit Test

- TIME and LOCATION: Laboratory class, week 12
- DURATION: 90 minutes
- TOPIC: Writing SQL to retrieve data. (week 7, 9 and10)
- PROVIDED MATERIAL:
  - –The data model (ER diagram).
  - –The schema (create table statements)
  - –The contents of each table.
  - – Database connection details.
- You will have access to Moodle and SQL Developer
- You may bring any additional printed material to the test, including your textbook (hard copy)

A database has been created for FIT2094 and permission granted for you to perform SELECT statements on the database. E.g.,
Select * from bugs.client;



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 UNI or BUGS account in Monash)

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

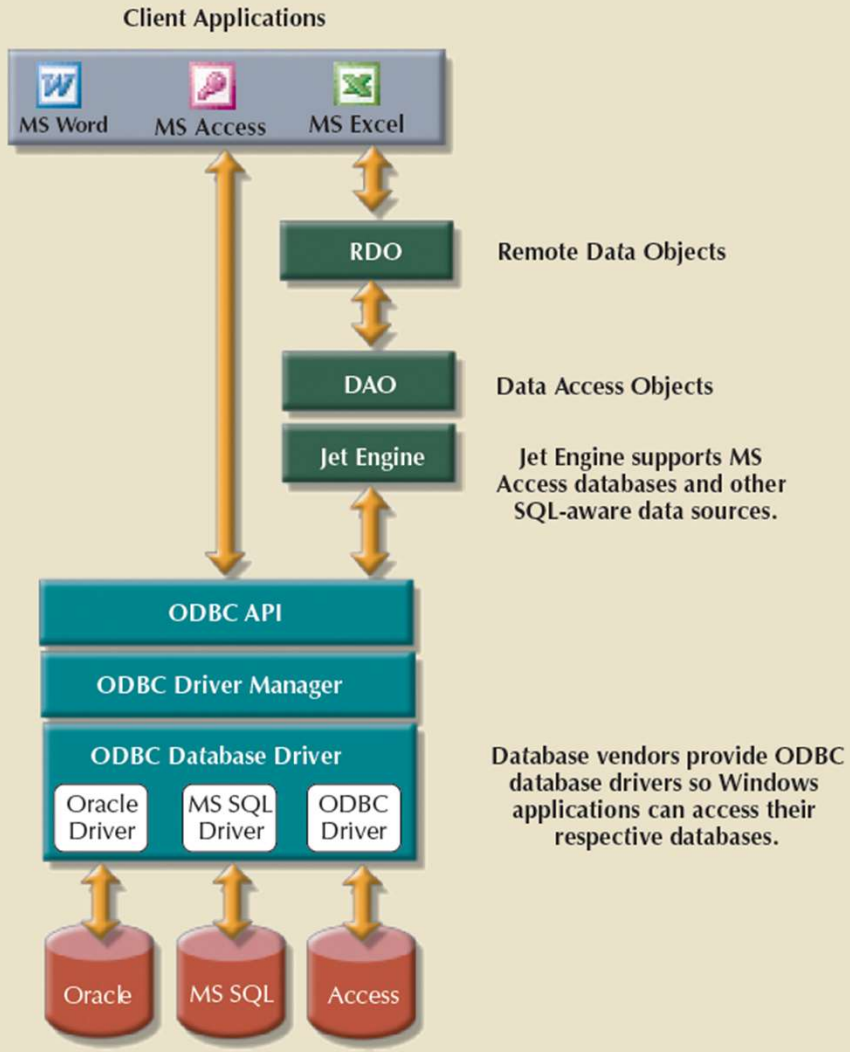
a. SQL
b. Database Middleware
c. The Data Layer
d. A Client Side Extension
e. Data Access Objects

Database Connectivity

Database Connectivity

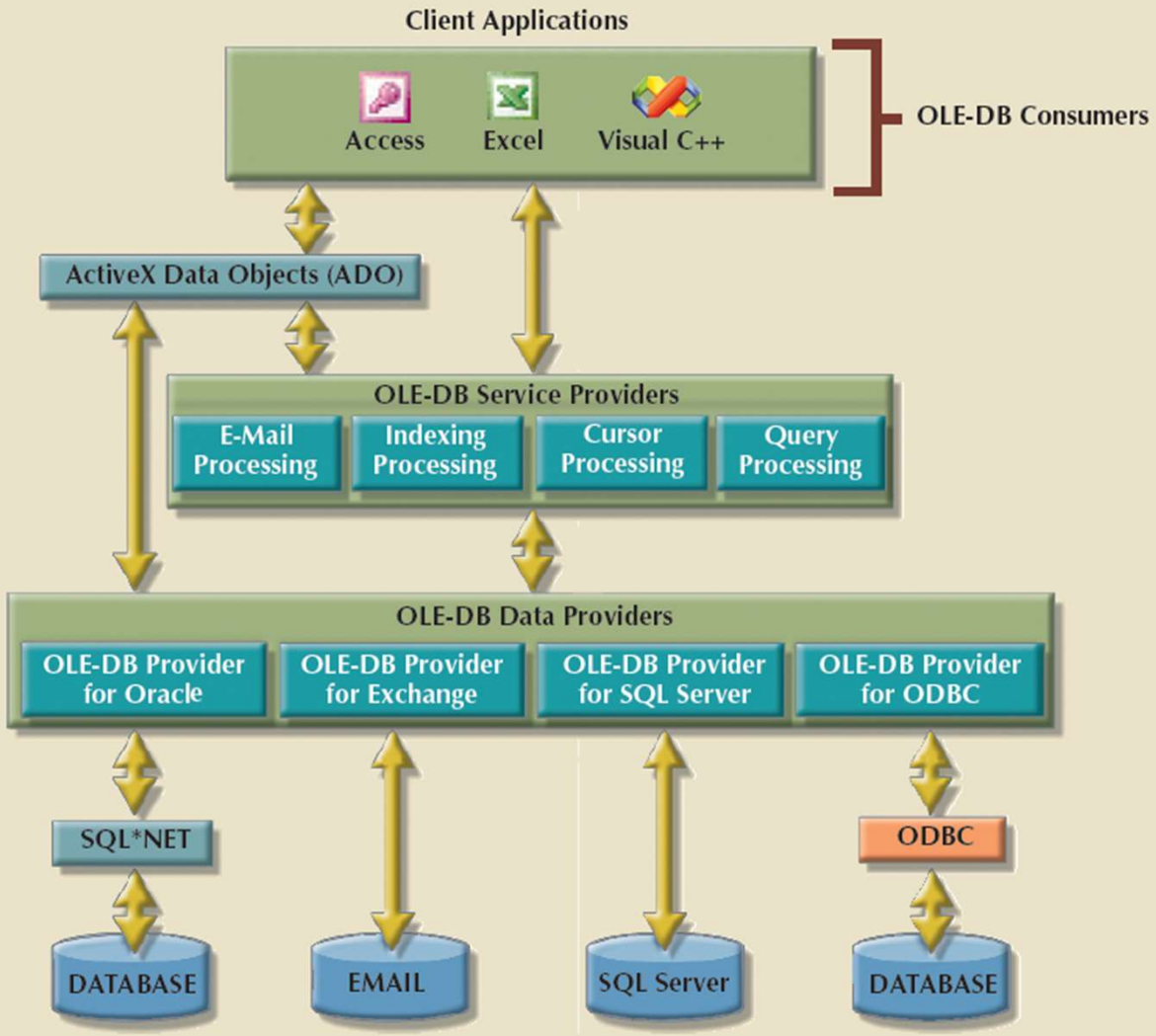
- The DATA LAYER – your data management application (DBMS)
- The DATABASE MIDDLEWARE – manages connectivity and data transformation issues. Many options available such as:
  - Native SQL Connectivity
    - Vendor provided eg. Oracle SQL*Net
  - Microsoft ODBC, DAO, RDO; OLE-DB and ADO.NET
  - Java Database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



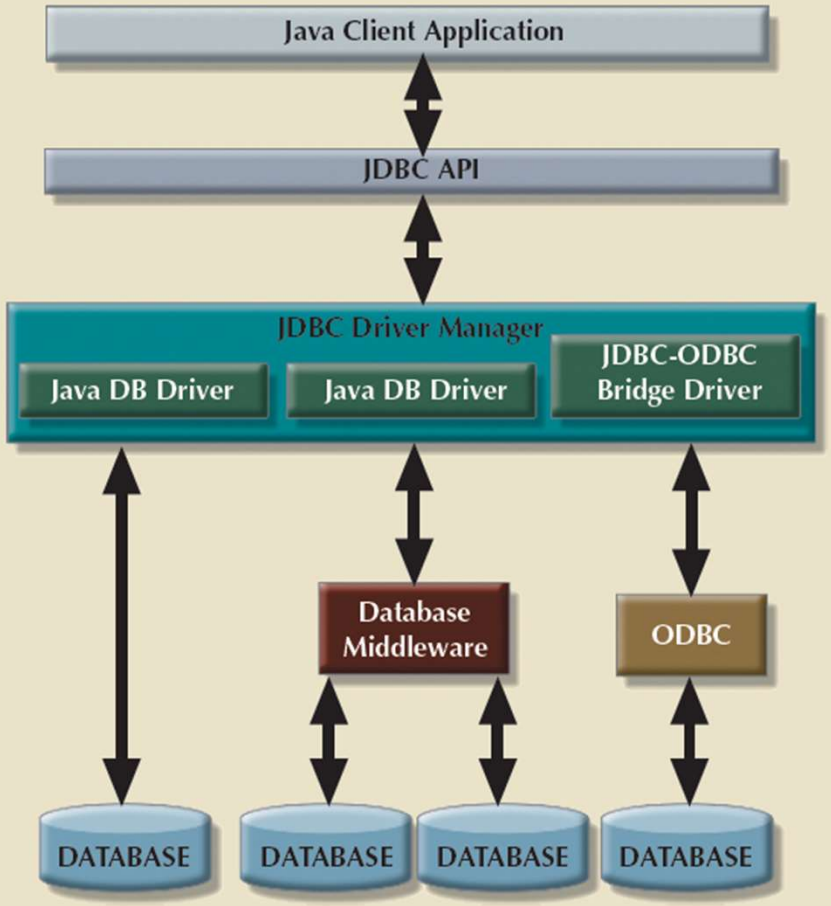
Coronel & Morris
Fig 15.2 Ed 12

FIGURE 15.5 OLE-DB ARCHITECTURE

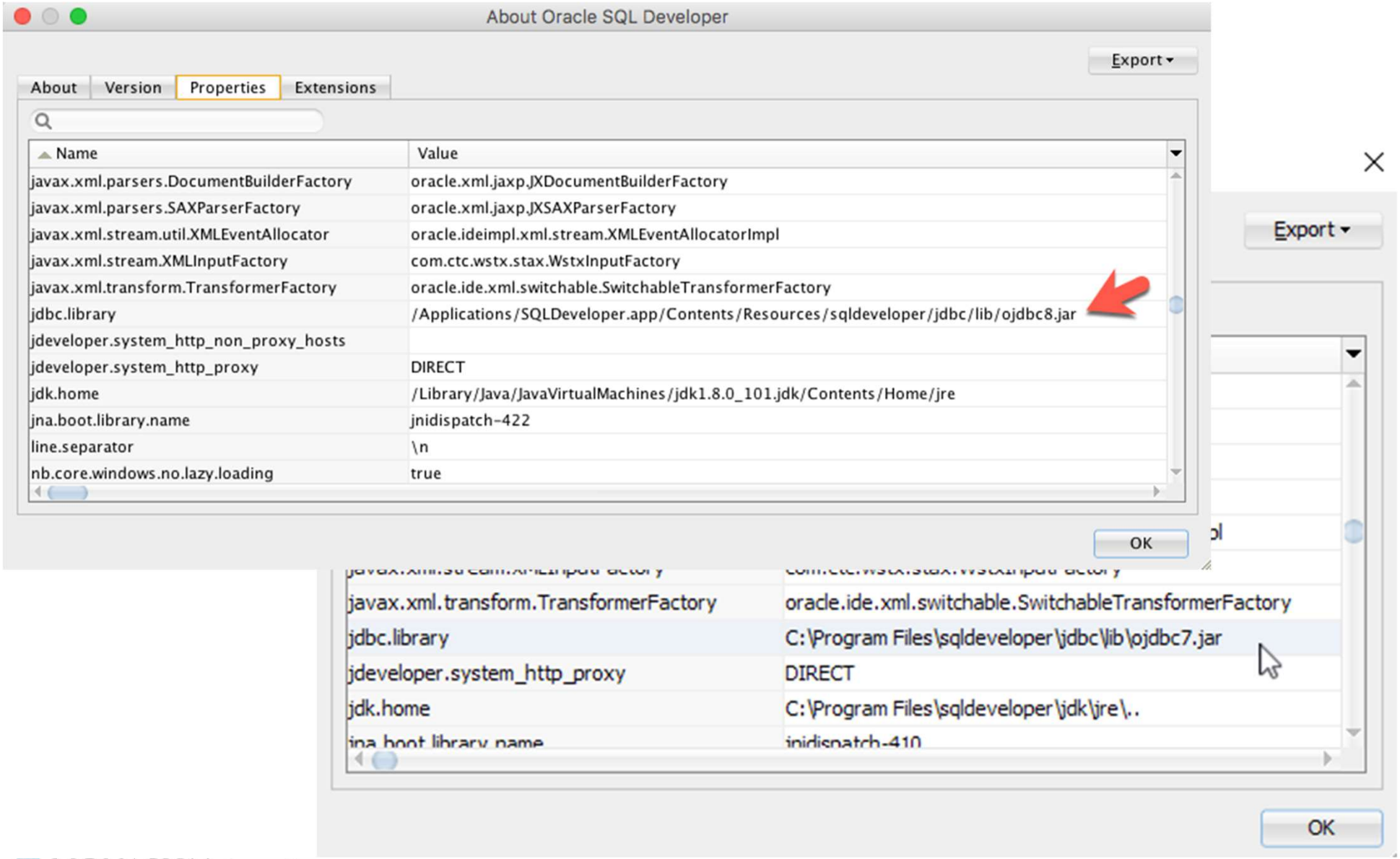


Coronel & Morris
Fig 15.5 Ed 12

FIGURE 15.7 JDBC ARCHITECTURE



Coronel & Morris
Fig 15.7 Ed 12


About Oracle SQL Developer
Export
About
Version
Properties
Extensions
Name
Value
javax.xml.parsers.DocumentBuilderFactory
oracle.xml.jaxp.JXDocumentBuilderFactory
javax.xml.parsers.SAXParserFactory
oracle.xml.jaxp.JXSAXParserFactory
javax.xml.stream.util.XMLEventAllocator
oracle.ideimpl.xml.stream.XMLEventAllocatorImpl
javax.xml.stream.XMLInputFactory
com.ctc.wstx.stax.WstxInputFactory
javax.xml.transform.TransformerFactory
oracle.ide.xml.switchable.SwitchableTransformerFactory
jdbc.library
/Applications/SQLDeveloper.app/Contents/Resources/sqldeveloper/jdbc/lib/ojdbc8.jar
jdeveloper.system_http_non_proxy_hosts
jdeveloper.system_http_proxy
DIRECT
jdk.home
/Library/Java/JavaVirtualMachines/jdk1.8.0_101.jdk/Contents/Home/jre
jna.boot.library.name
jnidispatch-422
line.separator
\n
nb.core.windows.no.lazy.loading
true
OK
Export
javax.xml.transform.TransformerFactory
oracle.ide.xml.switchable.SwitchableTransformerFactory
jdbc.library
C:\Program Files\sqldeveloper\jdbc\lib\ojdbc7.jar
jdeveloper.system_http_proxy
DIRECT
jdk.home
C:\Program Files\sqldeveloper\jdk\jre\..
OK

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/8I8R

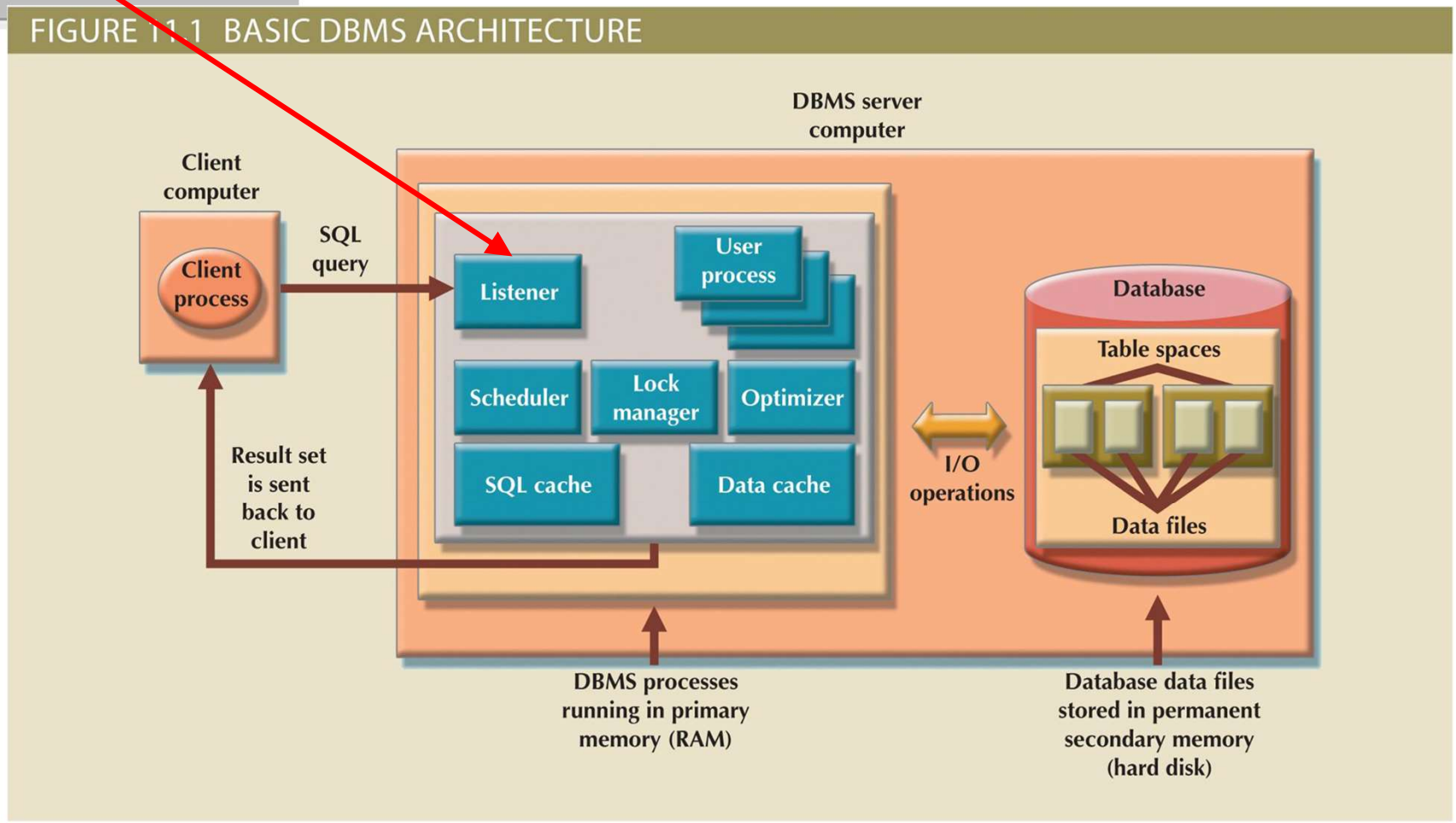
Placing application logic in the backend

- 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 available in 2018

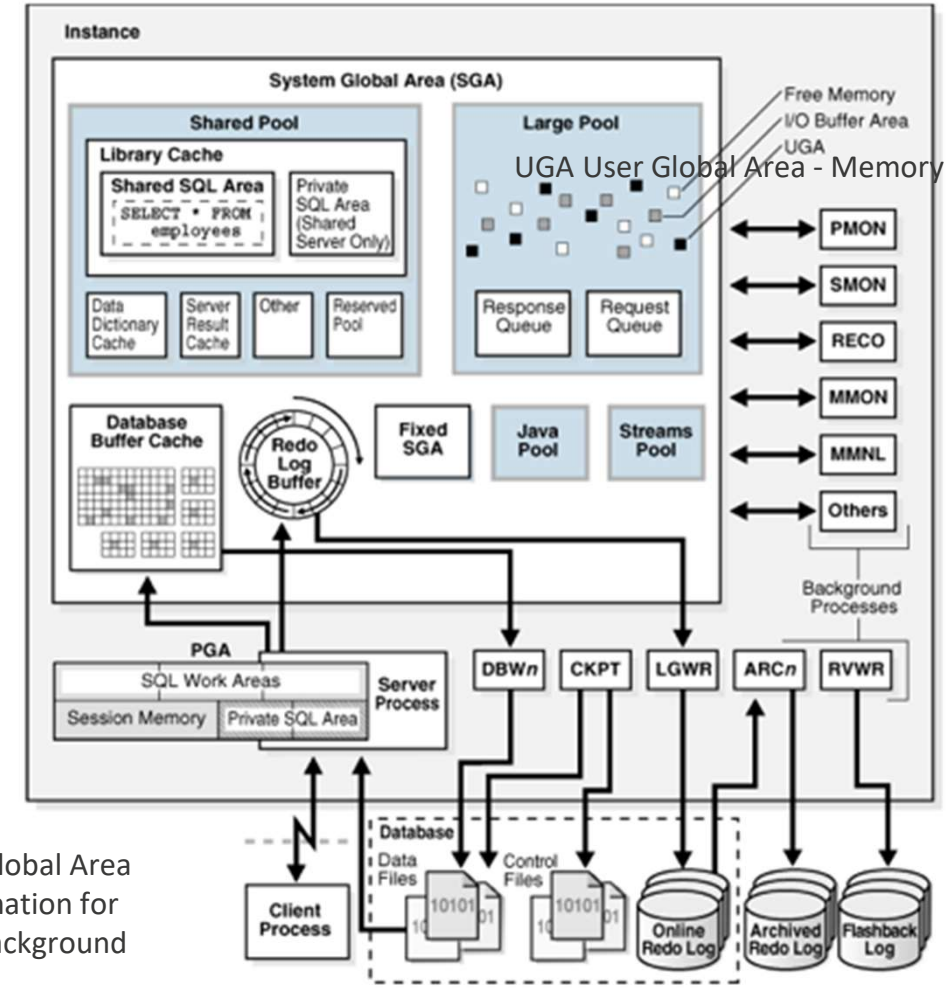
```
173     -- Procedure to add a new booking for a tour
174     PROCEDURE add_booking
175         (
176             arg_cust_no          IN book.cust_no%type,
177             arg_tour_no          IN book.tour_no%type,
178             arg_book_no_adults   IN book.book_no_adults%type,
179             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;
```

Oracle Architecture

Hostname hippo.its.monash.edu
Port 1521
SID FIT2094
Service name



Oracle Instance Architecture



UGA User Global Area - Memory associated with each user session eg. login info

https://goo.gl/o0hUJm

PGA - Program Global Area - data and information for each server or background process

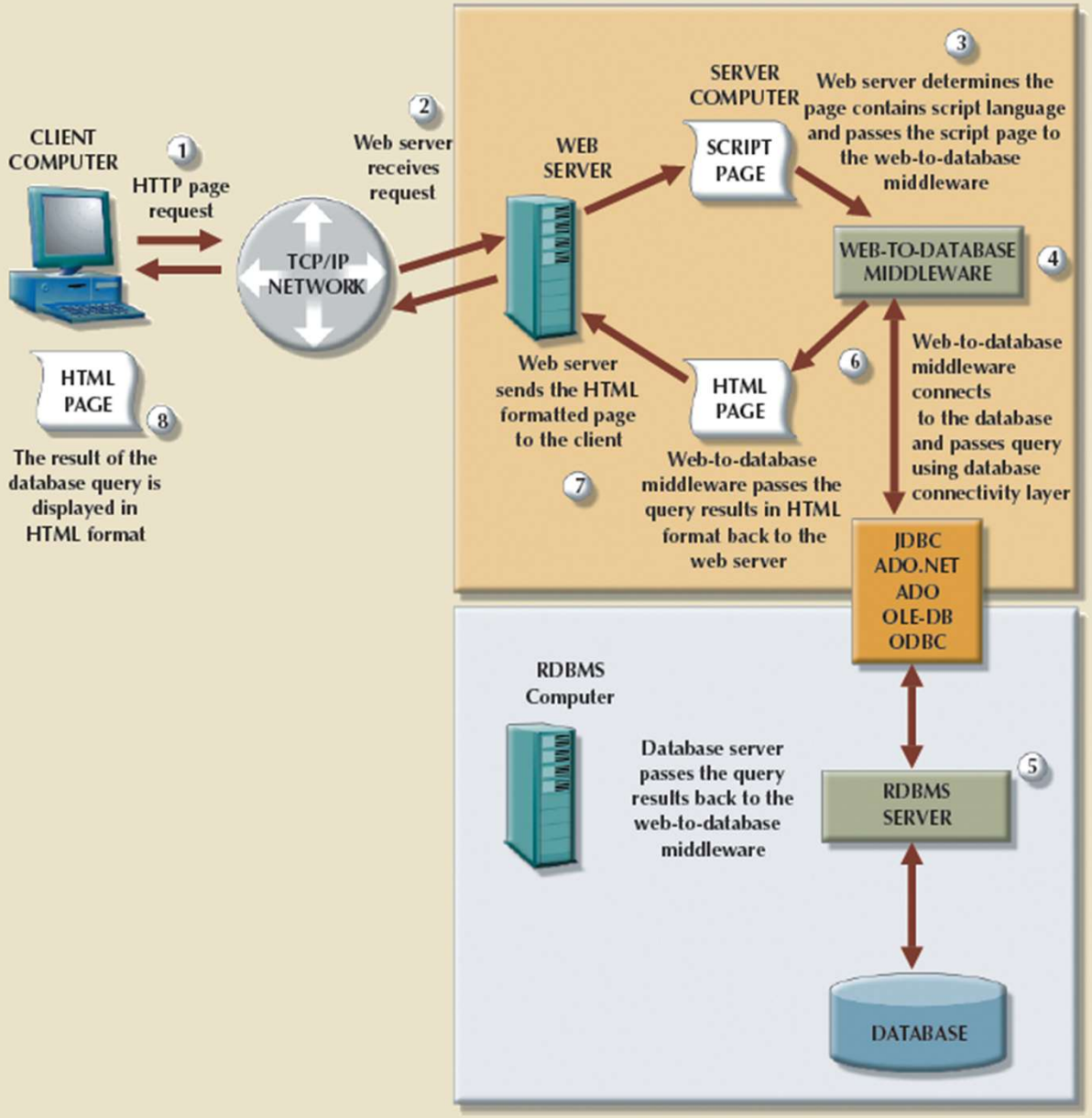
Databases on The Web

Database Internet Connectivity

☆ 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

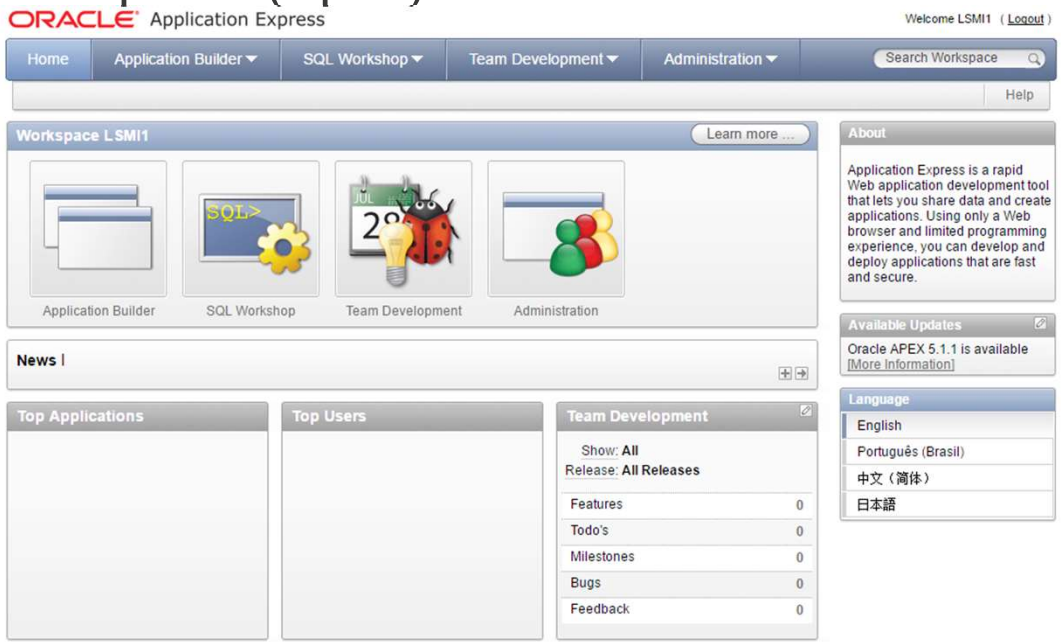
FIGURE 15.8 WEB-TO-DATABASE MIDDLEWARE



Coronel & Morris
Fig 15.8 Ed 12

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)



TIOBE Index for October 2017

Oct 2017	Oct 2016	Change	Programming Language	Ratings	Change
1	1		Java	12.431%	-6.37%
2	2		C	8.374%	-1.46%
3	3		C++	5.007%	-0.79%
4	4		C#	3.858%	-0.51%
5	5		Python	3.803%	+0.03%
6	6		JavaScript	3.010%	+0.26%
7	7		PHP	2.790%	+0.05%
8	8		Visual Basic .NET	2.735%	+0.08%
9	11	^	Assembly language	2.374%	+0.14%
10	13	^	Ruby	2.324%	+0.32%
11	15	^^	Delphi/Object Pascal	2.180%	+0.31%
12	9	v	Perl	1.963%	-0.53%
13	19	^^	MATLAB	1.880%	+0.26%
14	23	^^	Scratch	1.819%	+0.69%
15	18	^	R	1.684%	-0.06%

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

Unit Web Server and PHP

fit-db.infotech.monash.edu/

PHP Version 5.4.16

System	Linux corp-fit2094-v20.ocio.monash.edu 3.10.0-514.6.1.el7.x86_64 #1 SMP Sat Dec 10 11:15:38 EST 2016 x86_64
Build Date	Aug 5 2016 07:51:30
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

- Oracle interaction via Oracle OCI 8 functions (https://goo.gl/lR8Vi)
- **Set up login** details:

```
$dbUserName = "username";              <<< database user
$dbPassword = "whatever";              <<< database password

$dbInstance="connection string";   <<< database to connect to
```

  - Oracle Connection String ($dbInstance)

```
$dbInstance = "(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
    (HOST=fit2094.corp-prd.aws.monash.edu)(PORT=1521))
    (CONNECT_DATA=(SID=FIT2094)))";
```

PHP Database Access

- **Open database connection**, report error to browser and exit if not created:

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

  - Create HTML table header (output to be placed in a table) for browser to render

PHP Database Access

- **Build Query String** ($query)
- **Parse statement** (SQL select in $query string)

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

- **Execute the statement**

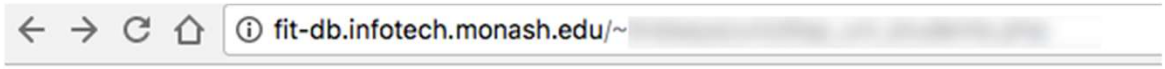
```
$r = oci_execute($stmt);
```

- **Fetch the results** of the Query

```
while (oci_fetch($stmt)) {
        ……
}
```

– generate HTML output for web page to be returned

- *See Week 11 Alexandria Lab exercise*

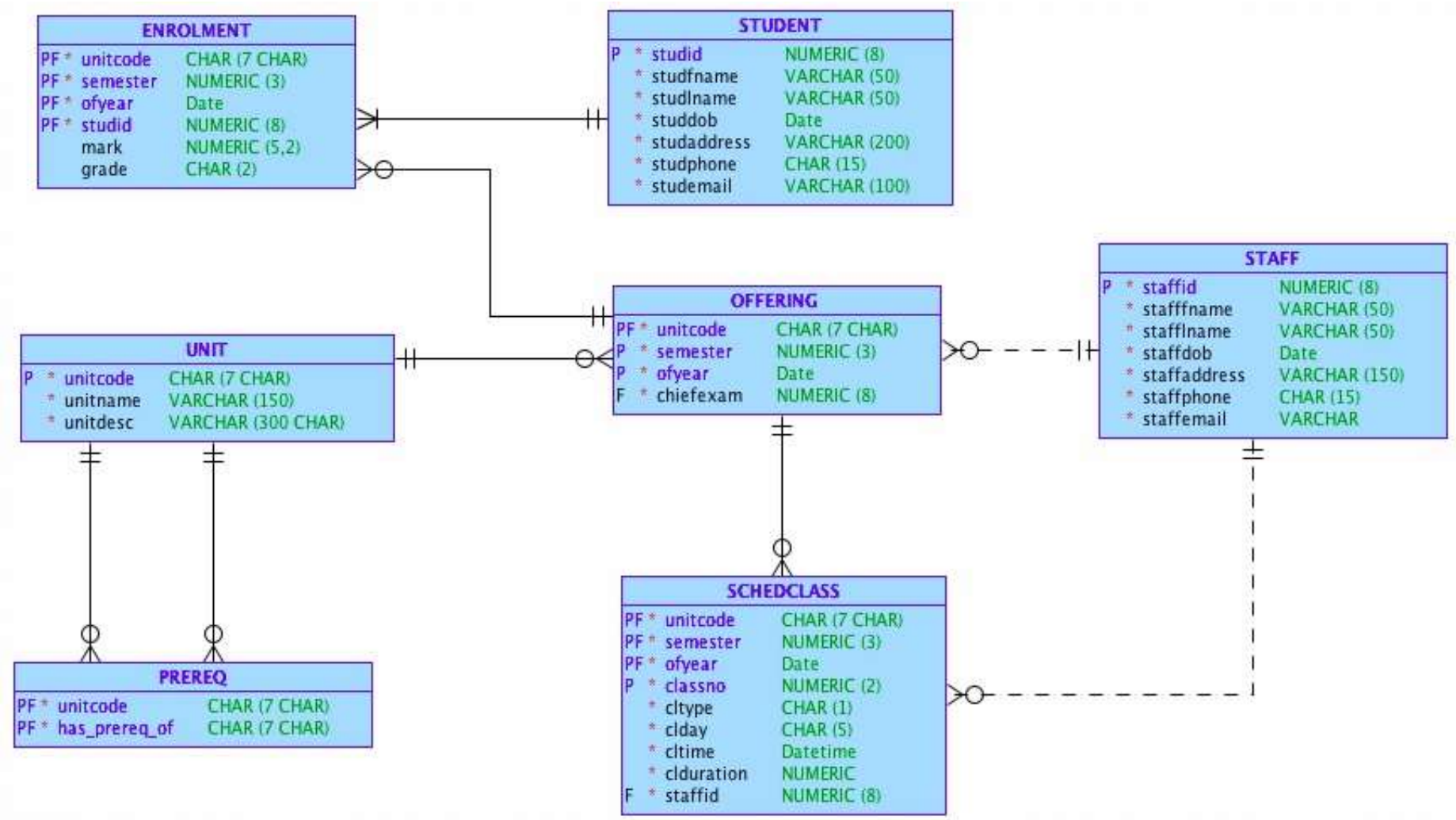


Student list UNIVERSITY database

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

Rows found:20

Practice



University data model