1Z0-061

Oracle Database 12c - SQL Fundamentals

Exam Summary – Syllabus – Questions

Table of Contents

Introduction to 1Z0-061 Exam on Oracle Database 12c - SQL Fundamentals

You can use this document to collect all the information about Oracle Database 12c - Installation and Administration (1Z0-062) certification. The Oracle 1Z0-062 certification is mainly targeted to those candidates who are from Database background and want to flourish their career with Oracle Database 12c Administrator Certified Associate (OCA) credential. The Oracle Database 12c - Installation and Administration certification exam validates your understanding of the Oracle Database technology and sets the stage for your future progression.

Oracle 1Z0-061 Certification Details:

Exam Name	Oracle Database 12c: Installation and Administration
Exam Code	1Z0-062
Exam Product Version	Oracle Database 12c
Exam Price	USD $245 (Pricing may vary by country or by localized currency)
Duration	150 Mins
Number of Questions	95
Passing Score	67%
Validated Against	This exam has been validated against Oracle Database 12.1.0.1.0.
Format	Multiple Choice
Recommended Training	Oracle Database 12c: Admin, Install and Upgrade Accelerated OR Oracle Database 12c: Install and Upgrade Workshop and Oracle Database 12c: Administration Workshop
Schedule Exam	Pearson VUE - Oracle
Recommended Practice	**1Z0-062 Online Practice Exam**

Oracle 1Z0-061 Exam Syllabus:

Subjects	Sub Topics
Exploring the Oracle Database Architecture	- List the architectural components of Oracle Database - Explain the memory structures - Describe the background processes - Explain the relationship between logical and physical storage structures
Oracle Database Management Tools	- Use database management tools
Oracle Database Instance	- Understand initialization parameter files - Start up and shut down an Oracle database instance - View the alert log and access dynamic performance views
Configuring the Oracle Network Environment	- Configure Oracle Net Services - Use tools for configuring and managing the Oracle network - Configure client-side network - Configure communication between databases
Managing Database Storage Structures	- Describe the storage of table row data in blocks - Create and manage tablespaces
Administering User Security	- Create and manage database user accounts - Grant and revoke privileges - Create and manage roles - Create and manage profiles
Managing Space	- Explain how Oracle database server automatically manages space - Save space by using compression - Proactively monitor and manage tablespace space usage - Use the Segment Advisor - Reclaim wasted space from tables and indexes by using the segment shrink functionality - Manage resumable space allocation
Managing Undo Data	- Explain DML and undo data generation - Monitor and administer undo data - Describe the difference between undo data and redo data - Configure undo retention
Managing Data Concurrency	- Describe the locking mechanism and how Oracle manages data concurrency - Monitor and resolve locking conflicts
Implementing Oracle Database Auditing	- Explain DBA responsibilities for security and auditing - Enable standard database auditing and unified auditing
Backup and Recovery Concepts	- Identify the importance of checkpoints, redo log files, and archive log files
Backup and Recovery Configuration	- Configure the fast recovery area - Configure ARCHIVELOG mode

Subjects	Sub Topics
Performing Database Backups	- Create consistent database backups - Back up your database without shutting it down - Create incremental backups - Automate database backups - Manage backups
Performing Database Recovery	- Determine the need for performing recovery - Use Recovery Manager (RMAN) and the Data Recovery Advisor to perform recovery of the control file, redo log file and data file
Moving Data	- Describe ways to move data - Use SQL*Loader to load data from a non-Oracle database - Use external tables to move data via platform-independent files - Explain the general architecture of Oracle Data Pump - Use Data Pump Export and Import to move data between Oracle databases
Performing Database Maintenance	- Manage the Automatic Workload Repository (AWR) - Use the Automatic Database Diagnostic Monitor (ADDM) - Describe and use the advisory framework - Set alert thresholds - User server-generated alerts - Use automated tasks
Managing Performance	- Use Automatic Memory Management - Use the Memory Advisor to size memory buffers
Managing Performance: SQL Tuning	- Manage optimizer statistics - Use the SQL Tuning advisor - Use the SQL Access Advisor to tune a workload
Managing Resources Using Database Resource Manager	- Configure the Database Resource Manager - Access and create resource plans - Monitor the Resource Manager
Automating Tasks by Using Oracle Scheduler	- Use Oracle Scheduler to simplify management tasks - Use job chains to perform a series of related tasks - Use Scheduler jobs on remote systems - Use advanced Scheduler features to prioritize jobs
Installing, Upgrading and Patching the Oracle Database	- Installing, Upgrading and Patching the Oracle Database
Oracle Software Installation Basics	- Plan for an Oracle Database software installation
Installing Oracle Grid Infrastructure for a Standalone Server	- Configure storage for Oracle Automatic Storage Management (ASM) - Install Oracle Grid Infrastructure for a standalone server
Installing Oracle Database Software	- Install the Oracle Database software

Subjects	Sub Topics
Creating an Oracle Database Using DBCA	- Create a database by using the Database Configuration Assistant (DBCA) - Generate database creation scripts by using DBCA - Manage database design templates by using DBCA - Configure database options by using DBCA
Using Oracle Restart	- Use Oracle Restart to manage components
Upgrading Oracle Database Software	- Describe upgrade methods - Describe data migration methods - Describe the upgrade process
Preparing to Upgrade to Oracle Database 12c	- Describe upgrade requirements when certain features or options are used in Oracle Database - Use the pre-upgrade information tool before performing an upgrade - Prepare the new Oracle home prior to performing an upgrade
Upgrading to Oracle Database 12c	- Upgrade the database to Oracle Database 12c by using the Database Upgrade Assistant (DBUA) - Perform a manual upgrade to Oracle Database 12c by using scripts and tools
Performing Post-Upgrade Tasks	- Migrate to unified auditing - Perform post-upgrade tasks
Migrating Data by Using Oracle Data Pump	- Migrate data by using Oracle Data Pump

1Z0-061 Sample Questions:

01) You query the database with this SQL statement:
SELECT AVG(LENGTH(name)) COLUMN1, SUM(INSTR(ssn,'52',2,2)) COLUMN2 FROM emp2 WHERE name = INITCAP (name);

Review the structure and data of the emp2 table.

Structure:

NAME	VARCHAR2(25)
SSN	VARCHAR2(9)

Data:

NAME	SSN
Joe	452852452
Mary	444525962
fred	445212525
Tom	492525252
Lucy	172826152

What will be displayed for the output of COLUMN1 and COLUMN2?
a) The value in COLUMN1 will be 3.5 and the value in COLUMN2 will be 11.

b) The value in COLUMN1 will be 3.6 and the value in COLUMN2 will be 19.
c) The value in COLUMN1 will be 3.5 and the value in COLUMN2 will be 26.
d) The value in COLUMN1 will be 3.6 and the value in COLUMN2 will be 26.
e) None of the above will be displayed for the output.

02) You query the database with this SQL statement:
SELECT id_number, NVL(100 / quantity, 0)
FROM product;

Which SQL SELECT statement capabilities are performed by this query?
a) selection only
b) projection only
c) selection and projection only
d) projection, selection, and joining

03) Which arithmetic expression will return a numeric value?
a) '14-FEB-2002' + 25
b) '03-JAN-2000' - 30
c) '17-JUN-1999' * (480/24)
d) TO_DATE('01-JAN-2001') - TO_DATE('01-DEC-2000')

04) Click the Exhibit(s) button to examine the data from the po_header and po_detail tables.

Examine the structures of the po_header and po_detail tables:
PO_HEADER

PO_NUM NUMBER NOT NULL
PO_DATE DATE DEFAULT SYSDATE
PO_TOTAL NUMBER(9,2)
SUPPLIER_ID NUMBER(9)
PO_TERMS VARCHAR2(25)

PO_DETATIL

PO_NUM NUMBER NOT NULL
PO_LINE_ID NUMBER NOT NULL
PRODUCT_ID NUMBER NOT NULL,
QUANTITY NUMBER(3) NOT NULL,
UNIT_PRICE NUMBER (5,2) DEFAULT 0,

The primary key of the po_header table is po_num. The primary key of the po_detail table is the combination of po_num and po_line_id. A foreign key constraint is defined on the po_num column of the po_detail table that references the po_header table.

You want to update the purchase order total amount for a given purchase order. The po_total column in the po_header table should equal the sum of the extended amounts of the corresponding po_detail records. You want the user to be prompted for the purchase order number when the query is executed.

When a purchase order is updated, the po_date column should be reset to the current date.

Which UPDATE statement should you execute?

a)
```
UPDATE po_header
SET po_total = (SELECT SUM(ext)
FROM (SELECT po_num, quantity * unit_price ext
FROM po_detail
WHERE po_num = &&ponum)),
SET po_date = sysdate
WHERE po_num = &&ponum;
```

b)
```
UPDATE po_header
SET po_total = (SELECT SUM(quantity * unit_price)
FROM (SELECT po_num)
FROM po_detail
WHERE po_num = &&ponum)),
po_date = DEFAULT
WHERE po_num = &&ponum;
```

c)
```
UPDATE po_header
SET po_total = (SELECT SUM(ext)
FROM (SELECT po_num, quantity * unit_price ext
FROM po_detail
WHERE po_num = &&ponum)),
UPDATE po_header
SET po_date = sysdate
WHERE po_num = &&ponum;
```

d)
```
UPDATE po_header
SET po_total = (SELECT SUM(ext)
FROM (SELECT po_num, quantity * unit_price ext
FROM po_detail
WHERE po_num = &&ponum)),
po_date = DEFAULT
WHERE po_num = &&ponum;
```

e)
```
UPDATE po_header
SET po_total = (SELECT po_num, SUM(ext)
FROM (SELECT po_num, quantity * unit_price ext
FROM po_detail
WHERE po_num = &&ponum)),
po_date = DEFAULT
WHERE po_num = &&ponum;
```

f)
```
UPDATE po_header
SET po_total = (SELECT SUM(ext)
FROM (SELECT po_num, quantity * unit_price ext
FROM po_detail
WHERE po_num = &&ponum)),
```

po_date = NULL
WHERE po_num = &&ponum;

05) The account table contains these columns:
ACCOUNT_ID NUMBER(12)
NEW_BALANCE NUMBER(7,2)
PREV_BALANCE NUMBER(7,2)
FINANCE_CHARGE NUMBER(7,2)

You need to create a single SELECT statement to accomplish these requirements:
- Display accounts that have a new balance that is less than the previous balance.
- Display accounts that have a finance charge that is less than $25.00.
- Display accounts that have no finance charge.

Evaluate this statement:
SELECT account_id
FROM account
WHERE new_balance < prev_balance
AND NVL(finance_charge, 0) < 25;

How many of the three requirements will this SELECT statement accomplish?
a) all of the requirements
b) one of the requirements
c) two of the requirements
d) none of the requirements

06) Examine the data in the product table.

PRODUCT

ID NUMBER	DESCRIPTION	MANUFACTURER ID	QUANTITY	COST
215	AAA 6pk-battery	NF10032	546	3.00
140	AA 2pk-battery	EL7968	2000	
603	D 2pk-battery	OT456	318	1.10
725	C 2pk-battery	OT456	239	.75
218	AAA 6pk-battery	OT456	980	3.15
220	AAA 8pk-battery	NF10032		4.20
126	AA 2pk-battery	NF10032	2513	
751	C 2pk-battery	EL7968	84	1.00

Evaluate this SELECT statement:
SELECT description, cost
FROM product
ORDER BY cost, quantity;

Which statements are true? (Choose all that apply.)
a) The product_id value for the first record displayed is 220.
b) The product_id values for the last two rows displayed are 140 and 126.
c) The description value for the first two rows displayed is C 2pk-battery.

d) The description value for the first two rows displayed is AA 2pk-battery.
e) No row with a product_id of 220is displayed.

07) Which two statements about the evaluation of clauses in a SELECT statement are true? (Choose two.)
a) The Oracle Server will evaluate a HAVING clause before a WHERE clause.
b) The Oracle Server will evaluate a WHERE clause before a GROUP BY clause.
c) The Oracle Server will evaluate a GROUP BY clause before a HAVING clause.
d) The Oracle Server will evaluate an ORDER BY clause before a WHERE clause.
e) The Oracle Server will evaluate an ORDER BY clause before a HAVING clause.

08) The employee table contains these columns:
EMPLOYEE_ID NUMBER NOT NULL
EMP_LNAME VARCHAR2(20) NOT NULL
EMP_FNAME VARCHAR2(10) NOT NULL
DEPT_ID NUMBER
SALARY NUMBER(9,2)

A user needs to retrieve information on employees who have the same department ID and salary as an employee ID that the user will enter. You want the query results to include employees who do not have a salary, but not the employee that the user entered.

Which statement will return the desired result?
a)
```
SELECT *
FROM employee
WHERE (department, salary) NOT IN
(SELECT department, salary)
FROM employee
WHERE employee_id = &1);
```
b)
```
SELECT *
FROM employee
WHERE (dept_id, salary) IN
(SELECT dept_id, NVL(salary, 0)
FROM employee
WHERE employee_id = &1);
```
c)
```
SELECT *
FROM employee
WHERE (dept_id, NVL(salary, 0)) IN
(SELECT dept_id, NVL(salary, 0)
FROM employee
WHERE employee_id = &&1)
AND employee_id <> &&1;
```
d)
```
SELECT *
FROM employee
WHERE (dept_id, salary) IN
(SELECT dept_id, salary)
```

```
FROM employee
WHERE employee_id = &1
AND salary IS NULL);
```

09) Click the Exhibit(s) button to examine the structures of the EMPLOYEE and TASK tables.

EMPLOYEE

EMPLOYEE_ID	NUMBER	NOT NULL, Primary Key
EMP_LNAME	VARCHAR2(25)	
EMP_FNAME	VARCHAR2(25)	
DEPT_ID	NUMBER	Foreign key to DEPT_ID column of DEPARTMENT table
JOB_ID	NUMBER	Foreign key to JOB_ID column of JOB table
MGR_ID	NUMBER	References EMPLOYEE_ID column
SALARY	NUMBER(9,2)	
HIRE_DATE	DATE	

TASK

PROJECT_ID	NUMBER	NOT NULL, Primary Key, Foreign key to PROJECT_ID column of PRODUCT table
TASK_ID	NUMBER	NOT NULL, Primary Key
TASK_DESCRIPTION	VARCHAR2(100)	
EST_COMPL_DATE	DATE	
EMPLOYEE_ID	NUMBER	Foreign key to EMPLOYEE_ID column of EMPLOYEE table

You need to produce a report containing all employees and all tasks. An employee must be included on the report even if he has no tasks assigned. All tasks, whether assigned to an employee or not, must also be included on the report.

Which SELECT statement should you use?

a)
```
SELECT e.emp_lname, e.emp_fname, t.task_description, t.est_compl_date
FROM employee e, task t
WHERE e.employee_id = t.employee_id;
```
b)
```
SELECT e.emp_lname, e.emp_fname, t.task_description, t.est_compl_date
FROM employee e, task t
WHERE e.employee_id (+) = t.employee_id;
```
c)
```
SELECT e.emp_lname, e.emp_fname, t.task_description, t.est_compl_date
FROM employee e, task t
WHERE e.employee_id = t.employee_id (+);
```
d)
```
SELECT e.emp_lname, e.emp_fname, t.task_description, t.est_compl_date
FROM employee e, task t
WHERE e.employee_id (+) = t.employee_id (+);
```
e) None of the options will produce the desired result.

10) Examine the structure of the employee table.

EMPLOYEE table (structure)

COLUMN_NAME	DATATYPE	CONSTRAINT
EMPLOYEE_ID	NUMBER	NOT NULL, Primary Key
EMP_LNAME	VARCHAR2(25)	
EMP_FNAME	VARCHAR2(25)	
DEPT_ID	NUMBER	Foreign key to DEPT_ID column of DEPARTMENT table
JOB_ID	NUMBER	Foreign key to JOB_ID column of JOB table
MGR_ID	NUMBER	Foreign Key to EMPLOYEE_ID column of EMPLOYEE table (this one)
SALARY	NUMBER(9,2)	
HIRE_DATE	DATE	

Which CREATE TABLE statement should you use to create the employee table?

a)

```
CREATE TABLE employee (
employee_id NUMBER,
emp_lname VARCHAR2(25) NOT NULL,
emp_fname VARCHAR2(25),
dept_id NUMBER,
job_id NUMBER,
mgr_id NUMBER,
salary NUMBER(9,2),
hire_date DATE,
CONSTRAINT employee_id_pk PRIMARY KEY(employee_id));
```

b)

```
CREATE TABLE employee (
employee_id NUMBER,
emp_lname VARCHAR2(25) NOT NULL,
emp_fname VARCHAR2(25),
dept_id NUMBER,
job_id NUMBER,
mgr_id NUMBER,
salary NUMBER(9,2),
hire_date DATE,
CONSTRAINT employee_id_pk PRIMARY KEY(employee_id),
CONSTRAINT mgr_id_fk FOREIGN KEY(mgr_id) REFERENCES employee(employee_id));
```

c)

```
CREATE TABLE employee (
employee_id NUMBER,
emp_lname VARCHAR2(25) NOT NULL,
emp_fname VARCHAR2(25),
dept_id NUMBER,
job_id NUMBER,
mgr_id NUMBER,
salary NUMBER(9,2),
hire_date DATE,
CONSTRAINT employee_id_pk PRIMARY KEY(employee_id),
CONSTRAINT dept_id_fk FOREIGN KEY(dept_id) REFERENCES department(dept_id),
CONSTRAINT job_id_fk FOREIGN KEY(job_id) REFERENCES job(job_id));
```

d)

```
CREATE TABLE employee (
employee_id NUMBER,
emp_lname VARCHAR2(25) NOT NULL,
emp_fname VARCHAR2(25),
dept_id NUMBER,
job_id NUMBER,
mgr_id NUMBER,
salary NUMBER(9,2),
hire_date DATE,
CONSTRAINT employee_id_pk PRIMARY KEY(employee_id),
CONSTRAINT dept_id_fk FOREIGN KEY(dept_id) REFERENCES department(dept_id),
CONSTRAINT job_id_fk FOREIGN KEY(job_id) REFERENCES job(job_id),
CONSTRAINT mgr_id_fk FOREIGN KEY(mgr_id) REFERENCES employee(employee_id));
```

Answers to 1Z0-061 Exam Questions:

QUESTION: 01 Answer: a	QUESTION: 02 Answer: b	QUESTION: 03 Answer: d	QUESTION: 04 Answer: d	QUESTION: 05 Answer: a
QUESTION: 06 Answer: b, c	QUESTION: 07 Answer: b, c	QUESTION: 08 Answer: c	QUESTION: 09 Answer: e	QUESTION: 10 Answer: d

Note: If you find any typo or data entry error in these sample questions, we request you to update us by commenting on this page or write an email on feedback@oraclestudy.com