



UNIVERSITÀ DEGLI STUDI DI MILANO

CONCORSO PUBBLICO, PER ESAMI, A N. 2 POSTI DI CATEGORIA C - AREA TECNICA, TECNICO-SCIENTIFICA ED ELABORAZIONE DATI, CON RAPPORTO DI LAVORO SUBORDINATO A TEMPO INDETERMINATO PRESSO L'UNIVERSITÀ DEGLI STUDI DI MILANO - DIREZIONE ICT - CODICE 22443

La Commissione giudicatrice della selezione, nominata con Determina Direttoriale n. 6615 del 02/05/2024, composta da:

Dott.ssa Loredana Rollandi	Presidente
Dott. Andrea Barbagallo	Componente
Dott. Alessandro Maggio	Componente
Sig.ra Susanna Emma Francesca Bianchi	Segretaria

comunica le tracce relative alla prova scritta.

TEMA n. 1

Numero totale di esercizi proposti: 4

Il candidato, utilizzando lo schema dati fornito di seguito (TABELLE e DATI), svolga i primi 3 esercizi proposti nelle pagine seguenti

TABELLE (script DDL in Oracle) – DATABASE SEMPLIFICATO GESTIONE ESAMI STUDENTI DI UN ATENEIO

```
CREATE table STUDENTE
(
  matricola          number(8) primary key,
  cognome            varchar2(50) not null,
  nome               varchar2(50) not null,
  data_nascita       date,
  luogo_nascita      varchar2(100),
  sesso              varchar2(1) not null
)
;

CREATE table INSEGNAMENTI
(
  codice_insegnamento  varchar2(6) primary key,
  descrizione            varchar2(100),
  lingua                varchar2(3) not null,
  crediti               number(2) not null,
  docente               varchar2(100),
  CdL                   varchar2(100)
)
;

CREATE table ESAMI_SOSTENUTI
(
  studente            number(8),
  insegnamento       varchar2(6),
  data                date not null,
  voto                number(2) not null,
  lode                 varchar2(1) not null,  ---- campo flag con valori S o N
)
```



UNIVERSITÀ DEGLI STUDI DI MILANO

primary key (studente, insegnamento),
foreign key (studente) references STUDENTE(matricola),
foreign key (insegnamento) references INSEGNAMENTI(codice_insegnamento)

)

;

DATI – DATABASE SEMPLIFICATO GESTIONE ESAMI STUDENTI DI UN ATENEO

TABELLA STUDENTE

MATRICOLA	COGNOME	NOME	DATA_NASCITA	LUOGO_NASCITA	SESSO
100	ROSSI	PAOLO	01/01/2000	MILANO	M
101	VERDI	DIEGO	15/02/2004	TORINO	M
102	BIANCHI	MARIA	20/03/2003	BARI	F
103	BALDINI	STEFANO	15/05/2002	MILANO	M
104	GORLA	SOFIA	20/03/2003	PAVIA	F
105	GATTI	MARIO	01/02/2002	PIACENZA	M
106	CIPOLLINI	MARIO	12/12/2001	BERGAMO	M
107	ROSSI	ADELE	03/03/2001	MILANO	F
108	BRIGNONE	SARA	23/11/2001	BERGAMO	F

TABELLA INSEGNAMENTI

CODICE_INSEGNAMENTO	DESCRIZIONE	LINGUA	DOCENTE	CDL	CREDITI
C02035	CHIMICA FISICA A	ITA	SIRONI PIERO	SCIENZE CHIMICHE	9
C03046	CHIMICA FISICA DEI MATERIALI	ITA	VECCHI PAOLA	SCIENZE CHIMICHE	6
F01023	ASTROFISICA TEORICA 1	ITA	BERTI PIERO	FISICA	6
F01037	FISICA DELLE PARTICELLE	ITA	SORANO GUIDO	FISICA	6
F02034	BIOFISICA	ITA	CONTE STEFANO	FISICA	6
M00001	ANALISI MATEMATICA 1	ITA	RAPETTI MARTA	MATEMATICA	9
M00003	LABORATORIO DI STATISTICA MATEMATICA	ITA	RIZZI PAOLO	MATEMATICA	3

TABELLA ESAMI_SOSTENUTI

STUDENTE	INSEGNAMENTO	DATA	VOTO	LODE
100	C02035	10/11/2023	27	N
101	M00001	10/10/2023	30	S
101	M00003	19/04/2024	24	N
106	C03046	11/10/2023	21	N
106	C02035	17/04/2024	30	N



ESERCIZIO 1

Scrivere una query SQL che per ogni INSEGNAMENTO della tabella INSEGNAMENTI estrae l'elenco degli STUDENTI che hanno superato l'esame con 30 con LODE. I campi colonna estratti dalla query devono essere: CODICE_INSEGNAMENTO, DESCRIZIONE, DOCENTE, CDL, MATRICOLA, COGNOME, NOME, DATA (dell'esame). L'elenco deve essere ordinato alfabeticamente in senso ascendente per codice_insegnamento e nominativo (cognome e nome) dello studente.

ESERCIZIO 2

Scrivere una vista SQL che per ogni STUDENTE calcola il valore medio dei VOTI ottenuti negli esami sostenuti e la somma dei CREDITI dei corrispondenti insegnamenti. La vista deve estrarre e mostrare i seguenti campi: MATRICOLA, COGNOME, NOME, la media dei VOTI ottenuti negli esami sostenuti, la somma dei corrispondenti CREDITI.

ESERCIZIO 3

Descrivere il funzionamento della seguente procedura PL/SQL di Oracle e l'output prodotto durante l'esecuzione:

```
CREATE OR REPLACE PROCEDURE STUDENT_PROCEDURE IS

CURSOR students_cursor IS
  SELECT MATRICOLA, NOME, COGNOME
  FROM STUDENTE
  WHERE MATRICOLA > 105;

v_matricola STUDENTE.MATRICOLA%TYPE;
v_nome STUDENTE.NOME%TYPE;
v_cognome STUDENTE.COGNOME%TYPE;

BEGIN

OPEN students_cursor;

LOOP

  FETCH students_cursor INTO v_matricola, v_nome, v_cognome;

  EXIT WHEN students_cursor%NOTFOUND;

  DBMS_OUTPUT.PUT_LINE(Nome: ' || v_nome || ' ' || v_cognome);

END LOOP;

CLOSE students_cursor;
END;
```

ESERCIZIO 4

Scrivere il codice Java per le due classi, *Fattura* e *RigaFattura*, che rappresentano rispettivamente le fatture e le righe delle fatture in un sistema di gestione delle fatture di acquisto di prodotti/servizi. La classe *Fattura* deve includere un metodo per calcolare il totale della fattura, sommando il valore di tutte le righe.

Attributi classe *Fattura*

- annoFattura (int): Anno di esercizio della fattura
- idFattura (int): Identificativo univoco della fattura.
- dataFattura (date): Data della fattura.
- righeFattura (List): Lista delle righe della fattura.

Attributi classe *RigaFattura*



- idRiga (int): Identificativo univoco della riga della fattura.
- idFattura (int): Identificativo della fattura a cui appartiene la riga
- descrizione (String): Descrizione del prodotto o servizio.
- importoRiga(double): Importo del prodotto o servizio

TEMA n. 2

Numero totale di esercizi proposti: 4

Il candidato, utilizzando lo schema dati fornito di seguito (TABELLE e DATI), svolga i primi 3 esercizi proposti nelle pagine seguenti

TABELLE (script DDL in Oracle) – DATABASE SEMPLIFICATO PER GESTIONE PARTECIPAZIONE A PROGETTI DEI DIPENDENTI DI UNA AZIENDA DI CONSULENZA IT

```
CREATE table IMPIEGATO
(
  matricola          number(5) primary key,
  cognome            varchar2(50) not null,
  nome               varchar2(50) not null,
  settore            varchar2(50) not null,
  stipendio_mese     number(8,2)
)
;

CREATE table PROGETTO
(
  codice_progetto    number(6) primary key,
  descrizione         varchar2(100),
  anno_inizio        number(4) not null,
  budget_progetto    number(12,2)
)
;

CREATE table PARTECIPAZIONE_PROGETTO
(
  impiegato          number(5),
  progetto           number(6),
  descrizione_ruolo  varchar2(100) not null, ---- nota su possibili valori:
                                     --- RESPONSABILE o PARTECIPANTE

  data_inizio        date,
  data_fine          date,

  primary key        (impiegato, progetto),
  foreign key (impiegato) references IMPIEGATO(matricola),
  foreign key (progetto) references PROGETTO(codice_progetto)
);
```



UNIVERSITÀ DEGLI STUDI DI MILANO

DATI – DATABASE SEMPLIFICATO PER GESTIONE PARTECIPAZIONE A PROGETTI DEI DIPENDENTI DI UNA AZIENDA DI CONSULENZA IT

TABELLA IMPIEGATO

MATRICOLA	COGNOME	NOME	SETTORE	STIPENDIO_MESE
100	FERLISI	MARIALUISA	ARCHITETTURE IT	3800
101	PERRI	MARIO	ARCHITETTURE IT	4200
102	FIORINI	SIMONE	ARCHITETTURE IT	4500
103	PESSINA	MARCO	SICUREZZA INFORMATICA	3000
104	RIZZI	CLAUDIO	SICUREZZA INFORMATICA	3800
105	GARLASCHI	SANDRO	SVILUPPO SOFTWARE	3100
106	MANCINI	LAURA	SVILUPPO SOFTWARE	3000
107	ROVIDA	PAOLO	SVILUPPO SOFTWARE	3200
108	VINCI	VALERIO	SVILUPPO SOFTWARE	4000

TABELLA PROGETTO

CODICE_PROGETTO	DESCRIZIONE	ANNO_INIZIO	BUDGET_PROGETTO
111111	IMPLEMENTAZIONE SISTEMA CYBERSECURITY ENTE CCC	2022	1200000
111112	IMPLEMENTAZIONE SOFTWARE GESTIONE PAGHE ENTE AAA	2022	1000000
111113	CONSULENZA CYBERSECURITY COMUNE BBB	2023	800000
111114	IMPLEMENTAZIONE SOFTWARE GESTIONE TIMBRATURE AZIENDA ZZZ	2023	700000
111115	CONSULENZA PROGETTAZIONE ARCHITETTURA IT AZIENDA XXX	2024	500000

TABELLA PARTECIPAZIONE_PROGETTO

IMPIEGATO	PROGETTO	DESCRIZIONE_RUOLO	DATA_INIZIO	DATA_FINE
100	111115	PARTECIPANTE	01/01/2024	
102	111115	RESPONSABILE	01/01/2024	
103	111111	PARTECIPANTE	01/02/2022	31/12/2023
104	111111	RESPONSABILE	01/02/2022	31/12/2023
104	111113	RESPONSABILE	01/09/2023	
105	111112	PARTECIPANTE	01/03/2022	31/12/2023
106	111112	PARTECIPANTE	01/03/2022	31/12/2023
106	111114	PARTECIPANTE	01/07/2023	
107	111114	RESPONSABILE	01/07/2023	
108	111112	RESPONSABILE	01/03/2022	31/12/2023



ESERCIZIO 1

Scrivere una query SQL che per ogni PROGETTO estrae l'elenco degli IMPIEGATI che stanno partecipando a progetti (progetti in corso). I campi estratti dalla query devono essere: CODICE_PROGETTO, DESCRIZIONE (del progetto), BUDGET_PROGETTO, MATRICOLA, COGNOME, NOME, DATA_INIZIO, DESCRIZIONE_RUOLO. L'elenco deve essere ordinato alfabeticamente in senso ascendente per descrizione del progetto e per nominativo (cognome e nome) dell'impiegato.

ESERCIZIO 2

Scrivere una vista SQL che per ogni IMPIEGATO estrae i seguenti campi e corrispondenti valori: MATRICOLA, COGNOME, NOME, SETTORE, il CONTEGGIO dei progetti a cui hanno partecipato o stanno partecipando con ruolo RESPONSABILE, il CONTEGGIO dei progetti a cui hanno partecipato o stanno partecipando con ruolo PARTECIPANTE.

ESERCIZIO 3

Descrivere il funzionamento della seguente procedura PL/SQL di Oracle e l'output prodotto durante l'esecuzione in base ai seguenti parametri di input:

- v_matricola = 200;
- v_matricola = 100;

```
CREATE OR REPLACE PROCEDURE EMPLOYEE_PROCEDURE(v_matricola IMPIEGATO.MATRICOLA%  
IS  
v_nome IMPIEGATO.NOME%TYPE;  
v_cognome IMPIEGATO.COGNOME%TYPE;  
BEGIN  
  
SELECT COGNOME, NOME  
INTO v_nome, v_cognome  
FROM IMPIEGATO  
WHERE MATRICOLA = v_matricola;  
  
DBMS_OUTPUT.PUT_LINE('Nome impiegato: ' || v_nome || ' ' || v_cognome);  
EXCEPTION  
WHEN NO_DATA_FOUND THEN  
DBMS_OUTPUT.PUT_LINE('Nessun impiegato trovato con matricola ' || v_matricola);  
WHEN OTHERS THEN  
DBMS_OUTPUT.PUT_LINE('Si è verificato un errore imprevisto: ' || SQLERRM);  
END;
```

ESERCIZIO 4

Scrivere il codice Java per le due classi, *Persona* e *ContattoPersona*, che rappresentano rispettivamente l'anagrafica e i contatti dell'anagrafica in un sistema di gestione delle anagrafiche del personale. La classe *Persona* deve includere un metodo che restituisca il numero totale dei contatti della Persona.

Classe *Persona*

- id (int): Identificativo univoco della persona.
- nome (String): Nome della persona.
- cognome (String): Cognome della persona.
- dataNascita (Date): Data di nascita della persona.
- contattoLista (List): Lista dei contatti della persona.

Classe *ContattoPersona*

- id (int): Identificativo univoco del contatto.
- tipoContatto (String): Tipologia di contatto (e.g., "EMPRV:email privata", "EMPEC: email PEC", "CEL:cellulare").



- contatto (String): Valore del contatto (e.g., l'indirizzo email o il numero di cellulare).

TEMA n. 3

Numero totale di esercizi proposti: 4

Il candidato, utilizzando lo schema dati fornito di seguito (TABELLE e DATI), svolga i primi 3 esercizi proposti nelle pagine seguenti

TABELLE (script DDL in Oracle) – DATABASE SEMPLIFICATO PER GESTIONE PRESTITO LIBRI IN UNA BIBLIOTECA UNIVERSITARIA

```
CREATE table STUDENTE
(
    matricola          number(8) primary key,
    cognome            varchar2(50) not null,
    nome               varchar2(50) not null,
    data_nascita       date,
    luogo_nascita      varchar2(100)
)
;

CREATE table LIBRO
(
    numInventario      varchar2(10) primary key,
    Titolo             varchar2 (100) NOT NULL,
    Lingua              varchar2 (20) NOT NULL,
    AnnoPubbl          Number(4) NOT NULL,
    Autore              Varchar2(100) NOT NULL
)
;

CREATE table PrendeInPrestito
(
    studente            number(8) NOT NULL,
    libro               varchar2(10) NOT NULL,
    data_inizio         date, --- nota: data inizio prestito
    data_fine           date, --- nota: data restituzione libro

    primary key (studente, libro),
    foreign key (studente) references STUDENTE (matricola),
    foreign key (libro) references LIBRO (numInventario)
)
;
```



UNIVERSITÀ DEGLI STUDI DI MILANO

DATI – DATABASE SEMPLIFICATO PER GESTIONE PRESTITO LIBRI IN UNA BIBLIOTECA UNIVERSITARIA TABELLA STUDENTE

MATRICOLA	COGNOME	NOME	DATA_NASCITA	LUOGO_NASCITA	SESSO
100	ROSSI	PAOLO	01/01/2000	MILANO	M
101	VERDI	DIEGO	15/02/2004	TORINO	M
102	BIANCHI	MARIA	20/03/2003	BARI	F
103	BALDINI	STEFANO	15/05/2002	MILANO	M
104	GORLA	SOFIA	20/03/2003	PAVIA	F
105	GATTI	MARIO	01/02/2002	PIACENZA	M
106	CIPOLLINI	MARIO	12/12/2001	BERGAMO	M
107	ROSSI	ADELE	03/03/2001	MILANO	F
108	BRIGNONE	SARA	23/11/2001	BERGAMO	F

TABELLA LIBRO

NUMINVENTARIO	TITOLO	LINGUA	ANNOPUBBL	AUTORE
AAA 001	TI PRENDO E TI PORTO VIA	ITALIANO	2000	NICOLO' AMMANITI
AAA 002	IO NON HO PAURA	ITALIANO	2001	NICOLO' AMMANITI
AAA 003	COME DIO COMANDA	ITALIANO	2015	NICOLO' AMMANITI
AAA 004	UN UOMO	ITALIANO	2014	ORIANA FALLACI
AAA 005	L'AMICA GENIALE	ITALIANO	2011	ELENA FERRANTE
BBB 001	IL NOME DELLA ROSA	ITALIANO	1980	UMBERTO ECO
BBB 002	SE QUESTO E' UN UOMO	ITALIANO	1947	PRIMO LEVI
BBB 003	L'ISOLA DI ARTURO	ITALIANO	1957	ELSA MORANTE
BBB 004	IL BARONE RAMPANTE	ITALIANO	1957	ITALO CALVINO
CCC 001	ULYSSES	INGLESE	1922	JAMES JOYCE
CCC 002	DUBLINERS	INGLESE	1914	JAMES JOYCE
CCC 003	EN ATTENDANDT GODOT	INGLESE	1952	SAMUEL BECKETT

TABELLA PRENDEINPRESTITO

STUDENTE	LIBRO	DATA_INIZIO	DATA_FINE
100	AAA 001	01/03/2024	
100	BBB 001	01/02/2023	31/05/2023
100	CCC 001	01/09/2023	30/11/2023
102	AAA 002	01/04/2022	31/07/2022
102	BBB 002	01/10/2022	30/11/2022
104	CCC 003	01/02/2024	
104	BBB 002	01/02/2023	31/05/2023
106	AAA 003	01/03/2024	
106	AAA 005	01/03/2024	
108	BBB 004	01/03/2021	30/06/2021



ESERCIZIO 1

Scrivere una query SQL che per ogni LIBRO estrae l'elenco degli studenti che hanno preso il libro in prestito (in passato o ad oggi). I campi estratti dalla query devono essere: NUMINVENTARIO, TITOLO, ANNOPUBBL, AUTORE, MATRICOLA, COGNOME, NOME, DATA_INIZIO, DATA_FINE. L'elenco deve essere ordinato alfabeticamente in senso ascendente per NUMIVENTARIO e nominativo (cognome e nome) dello studente.

ESERCIZIO 2

Scrivere una vista SQL che per ogni STUDENTE estrae i seguenti campi e corrispondenti valori: MATRICOLA, COGNOME, NOME, il conteggio dei LIBRI in lingua italiana presi in prestito, il conteggio dei LIBRI in lingua inglese presi in prestito (nota: sia prestiti passati che prestiti in corso).

ESERCIZIO 3

Descrivere il funzionamento della seguente procedura PL/SQL di Oracle e l'output prodotto durante l'esecuzione:

```
CREATE OR REPLACE PROCEDURE BOOK_PROCEDURE IS

v_count number;

BEGIN

INSERT INTO LIBRO(NUMINVENTARIO, TITOLO, LINGUA, ANNOPUBBL, AUTORE)
VALUES('ZZZ 001', 'IL GATTOPARDO', 'ITALIANO', 1958, 'GIUSEPPE TOMASI DI LAMPEDUSA');

COMMIT;

INSERT INTO LIBRO(NUMINVENTARIO, TITOLO, LINGUA, ANNOPUBBL, AUTORE)
VALUES('ZZZ 002', 'IL PICCOLO PRINCIPE', 'FRANCESE', 1943, 'ANTOINE DE SAINT-EXUPÉRY');

UPDATE LIBRO SET ANNOPUBBL = 2024;

SELECT COUNT(*) INTO v_count FROM LIBRO;

DBMS_OUTPUT.PUT_LINE('Numero record: ' || v_count);

ROLLBACK;

END;
```

ESERCIZIO 4

Scrivere il codice Java per le due classi, *Ordine* e *RigaOrdine*, che rappresentano rispettivamente gli ordini e le righe degli ordini in un sistema di gestione del ciclo degli acquisti. La classe *Ordine* deve includere un metodo per calcolare il totale dell'ordine, sommando il valore di tutte le righe.

Classe *Ordine*

- idOrdine (int): Identificativo univoco dell'ordine.
- annoOrdine (int): Anno di esercizio dell'ordine
- dataOrdine (Date): Data dell'ordine.
- righeOrdine (List): Lista delle righe dell'ordine.

Classe *RigaOrdine*

- idRiga (int): Identificativo univoco della riga dell'ordine.
- idOrdine (int): Identificativo dell'ordine a cui appartiene la riga.
- descrizione (String): Descrizione del prodotto o servizio.



UNIVERSITÀ DEGLI STUDI DI MILANO

- importoRiga (double): Importo totale del prodotto o servizio.

Milano, 27 maggio 2024

La Commissione

Dott.ssa Loredana Rollandi Presidente

Dott. Andrea Barbagallo Componente

Dott. Alessandro Maggio Componente

Sig.ra Susanna Emma Francesca Bianchi Segretaria