



# UNIVERSITÀ DEGLI STUDI DI MILANO

CONCORSO PUBBLICO, PER TITOLI ED ESAMI, A N. 1 POSTO DI CATEGORIA D - POSIZIONE ECONOMICA D1 - AREA TECNICA, TECNICO-SCIENTIFICA ED ELABORAZIONE DATI, SISTEMISTA SULL'INFRASTRUTTURA DI INTEGRAZIONE (MIDDLEWARE), CON RAPPORTO DI LAVORO SUBORDINATO A TEMPO INDETERMINATO, PRESSO LA DIREZIONE SISTEMI INFORMATIVI D'ATENEO - BANDITO CON DETERMINA N. 14489/2018 DEL 16.10.2018 E PUBBLICATO SULLA G.U. N. 91 DEL 16.11.2018 - CODICE 19479

La Commissione giudicatrice del concorso, nominata con determina n. 18672/2018 del 18.12.2018 e così composta:

DOTT. SALERNO ANTONIO - PRESIDENTE

DOTT.SSA ARONA ARIANNA - COMPONENTE

DOTT. PUPILLO VINCENZO - COMPONENTE

SIG. FICHERA FRANCESCO - SEGRETARIO

comunica le tracce relative alla prima prova scritta:

## TRACCIA n. 1

Il **Middleware** è definito come un insieme di servizi di supporto alla distribuzione delle informazioni, indipendenti dalle applicazioni. Si tratta del software che risiede al di sopra del sistema operativo ed al di sotto delle applicazioni. Nell'ambito delle integrazioni applicative eterogenee particolare rilevanza assume l'**Enterprise Service Bus**. Il candidato descriva le caratteristiche tipiche di un **Enterprise Service Bus** e come possono essere impiegate.

## TRACCIA n. 2

Il **Web Services** si sono affermati principalmente nello sviluppo di applicazioni web. Il candidato descriva cosa sono i **Web Services** basati sul protocollo **SOAP**, quali sono le loro caratteristiche principali e i vantaggi/svantaggi rispetto ai **Web Services RESTfull**.

## TRACCIA n. 3

Le architetture di **Middleware** si sono dimostrate particolarmente rilevanti quando si parla di sistemi distribuiti, in quanto esse sono in grado di nascondere l'eterogeneità dei devices e offrire interfacce e risorse che appaiono omogenee ai layers applicativi. In questo scenario, negli ultimi anni, svolge un ruolo di primaria importanza lo stile di architettura del software denominato **REST**. Il candidato descriva le caratteristiche dei **Web Services** conformi a questo stile (denominati **RESTfull webservice**).

La Commissione comunica le tracce relative alla seconda prova scritta a contenuto teorico-pratico:

## TRACCIA n. 1

### Prova pratica n. 1

#### Domanda 1

Facendo riferimento alla struttura tabellare sotto riportata, progettare una semplice api **RESTfull**, con output in **JSON**, che consenta di consultare le informazioni sui registri dei docenti (le due tabelle fornite).



L'API deve consentire di cercare uno singolo registro, o di ottenere una lista paginata di registri filtrabile per data.

Della API bisogna definire gli endpoint/s del servizio/servizi con i relativi path-params e/o query-params, motivandone le scelte.

Per quanto riguarda l'output in JSON: definire la struttura del messaggio (astraendosi, per quanto possibile, dalla struttura dati), motivando le scelte fatte.

Cosa restituisce il servizio in caso di mancanza del dato richiesto? E in caso di errore?

## \* Struttura dati

I docenti, per ogni insegnamento (esempio: Analisi 1, Programmazione 1 etc), sono tenuti a compilare un registro delle attività svolte nei vari giorni di lezione.

Qui di seguito sono riportate due tabelle in relazione 1:N

### TABELLA REGISTRO

ID	NOT NULL VARCHAR(64)
ANAD_COD_FISCALE	NOT NULL CHAR(16)
ANNO_ACC	NUMBER(4)
CODFAC_DOC	VARCHAR2(10)
CODICE_CORSO_DI_LAUREA	VARCHAR2(20)
CODICE_INSEGNAMENTO	VARCHAR2(20)
DATA_CONFERMA	DATE
DATA_CHIUSURA	DATE
STATO	VARCHAR2(20) -- (può essere Aperto, Chiuso, Confermato)
ANNOTAZIONI	VARCHAR2(4000)
DATA_CREAZIONE	DATE
DATA_UPDATE	DATE

### TABELLA ATTIVITA

ID	NOT NULL VARCHAR(64)
ID_REGISTRO	VARCHAR(64)
DATA_ATTIVITA	DATE
DURATA	NUMBER(5,1)
AULA	VARCHAR2(40)
SEDE	VARCHAR2(256)
ARGOMENTO	VARCHAR2(5000)
DATA_CREAZIONE	DATE
DATA_UPDATE	DATE

## Domanda 2

Considerato lo schema E/R e i dati riportati nell'allegato 1, rispondere alle seguenti domande:

1. Descrivere le entità ROLE e PROJECT\_PARTICIPATION dello schema ER allegato, indicando la cardinalità delle relazioni esistenti; definire lo statement DDL per l'aggiunta all'entità ROLE di un attributo POSITION non nullo per l'introduzione di SENIOR o JUNIOR;
2. Per ogni dipartimento calcolare il totale salario percepito da tutti i dipendenti esclusi i MANAGER. La query deve esporre gli attributi sotto indicati per i soli dipartimenti con totale salario maggiore di 8.000\$:

a. Num. del dipartimento, Nome del dipartimento, Salario totale



### Domanda 3

Produrre uno script bash che:

- esegue lo script "coperture.sh" che si trova nella home dell'utente myUser utilizzando i diritti dell'utente myUser (risultato dell'esecuzione riportato in calce);
- dal risultato dell'elaborazione, scritto nella stessa directory, estrae la data evidenziata e la confronta con la data attuale;
- se la data attuale è successiva a quella contenuta nel file copia il file sulla macchina remote.node utilizzando i permessi dell'utente myUser.

### Risultato dell'esecuzione

```
<exist:result xmlns:exist="http://exist.sourceforge.net/NS/exist">
  <exist:collection name="/db/facolta_1E"
created="2018-06-05T15:15:49.071+02:00" owner="admin" group="dba"
permissions="rwur-ur-u">
  <exist:resource name="docenti"
created="2018-08-10T02:45:25.547+02:00"
last-modified="2018-08-10T02:45:25.547+02:00" owner="admin" group="dba"
permissions="rwur-ur-u"/>
  <exist:resource name="coperture"
created="2018-08-10T02:45:32.849+02:00"
last-modified="2018-08-10T02:45:32.849+02:00" owner="admin" group="dba"
permissions="rwur-ur-u"/>
  <exist:resource name="facolta"
created="2018-06-05T15:26:01.223+02:00"
last-modified="2018-08-10T02:48:56.62+02:00" owner="admin" group="dba"
permissions="rwur-ur-u"/>
  <exist:resource name="strutture"
created="2018-08-10T02:49:38.343+02:00"
last-modified="2018-08-10T02:49:38.343+02:00" owner="admin" group="dba"
permissions="rwur-ur-u"/>
  <exist:resource name="search_data"
created="2018-08-10T02:49:47.887+02:00"
last-modified="2018-08-10T02:49:47.887+02:00" owner="admin" group="dba"
permissions="rwur-ur-u"/>
</exist:collection>
</exist:result>
```

TRACCIA n. 2

### Prova pratica n. 2

#### Domanda 1

Progettare una semplice api RESTfull, con output in JSON, che consenta di consultare le informazioni sui registri dei docenti (le due tabelle fornite).

L'API deve consentire di cercare un singolo registro, un insieme di registri o l'intero insieme dei registri, filtrabile per stato, con o senza i dati provenienti dalle tabelle associate.



Della API bisogna definire gli endpoint/s del servizio/servizi con i relativi path-params e/o query-params, motivandone le scelte.

Per quanto riguarda l'output in JSON: definire la struttura del messaggio (astraendosi, per quanto possibile, dalla struttura dati), motivando le scelte fatte.

Cosa restituisce il servizio in caso di mancanza del dato richiesto? E in caso di errore?

## \* Struttura dati

I docenti, per ogni insegnamento (esempio: Analisi 1, Programmazione 1 etc), sono tenuti a compilare un registro delle attività svolte nei vari giorni di lezione.

Qui di seguito sono riportate due tabelle in relazione 1:N

### TABELLA REGISTRO

ID	NOT NULL VARCHAR(64)
ANAD_COD_FISCALE	NOT NULL CHAR(16)
ANNO_ACC	NUMBER(4)
CODFAC_DOC	VARCHAR2(10)
CODICE_CORSO_DI_LAUREA	VARCHAR2(20)
CODICE_INSEGNAMENTO	VARCHAR2(20)
DATA_CONFERMA	DATE
DATA_CHIUSURA	DATE
STATO	VARCHAR2(20) -- (può essere Aperto, Chiuso, Confermato)
ANNOTAZIONI	VARCHAR2(4000)
DATA_CREAZIONE	DATE
DATA_UPDATE	DATE

### TABELLA ATTIVITA

ID	NOT NULL VARCHAR(64)
ID_REGISTRO	VARCHAR(64)
DATA_ATTIVITA	DATE
DURATA	NUMBER(5,1)
AULA	VARCHAR2(40)
SEDE	VARCHAR2(256)
ARGOMENTO	VARCHAR2(5000)
DATA_CREAZIONE	DATE
DATA_UPDATE	DATE



## Domanda 2

Considerato lo schema E/R e i dati riportati nell'allegato 1, rispondere alle seguenti domande:

1. Descrivere le entità PROJECT e PROJECT\_PARTICIPATION dello schema ER allegato, indicando la cardinalità delle relazioni esistenti; definire lo statement DDL per l'aggiunta all'entità PROJECT di un attributo BUDGET non nullo.
3. Calcolare per ogni dipartimento il salario minimo, massimo e la media erogati. La query deve esporre gli attributi sotto indicati per i soli dipartimenti che totalizzano un salario massimo compreso tra 3.000\$ e 5.000\$:
  - a. Num. del dipartimento, Nome del dipartimento, Sal\_Min, Sal\_Max, Sal\_Avg

## Domanda 3

Descrivere lo script sotto riportato:

Descrivere lo script sotto riportato:

```
#!/bin/bash
```

```
WHOAMI=`/usr/bin/whoami`
if [ "${WHOAMI}" != "authUser" ];then
    echo "ERROR: Utente non abilitato alla esecuzione dello script"
    exit 1
fi

server=remote.node

for myVar in me you;
do

    if [ ! -f files/RSYNC_http_exclude ]; then
        OPTIONS_EXCLUDE="--exclude-from=files_fare/RSYNC_http_exclude_${myVar}"
    fi

    SOURCE="/myPath/${myVar}/"
    DESTINATION="remoteUser@${server}:/remotePath/${myVar}/htdocs"

    case $myVar in
        me) OPTIONS="${OPTIONS_EXCLUDE} -e ssh -a";;
        you) SOURCE="/myPath/" OPTIONS="${OPTIONS_EXCLUDE} -e ssh -a";;
    esac

    CMD=`ps -ef | grep rsync | grep httpd |grep $myVar`
    PID=${CMD}
    if [ -n "$PID" ]; then
        /usr/bin/kill $PID
    else
        rsync ${OPTIONS} ${SOURCE} ${DESTINATION}
    fi

done
```



TRACCIA n. 3

## Prova pratica n. 3

### Domanda 1

Progettare una semplice api RESTfull, con output in JSON, che consenta di consultare le informazioni sui registri dei docenti (le due tabelle fornite).

L'API deve consentire di cercare tutte le attività in un range di giornate (dal .. al ..), con o senza i dati dei registri eventualmente associati.

Della API bisogna definire gli endpoint/s del servizio/servizi con i relativi path-params e/o query-params, motivandone le scelte.

Per quanto riguarda l'output in JSON: definire la struttura del messaggio (astraendosi, per quanto possibile, dalla struttura dati), motivando le scelte fatte.

Cosa restituisce il servizio in caso di mancanza del dato richiesto? E in caso di errore?

### \* Struttura dati

I docenti, per ogni insegnamento (esempio: Analisi 1, Programmazione 1 etc), sono tenuti a compilare un registro delle attività svolte nei vari giorni di lezione.

Qui di seguito sono riportate due tabelle in relazione 1:N

#### TABELLA REGISTRO

ID	NOT NULL VARCHAR(64)
ANAD_COD_FISCALE	NOT NULL CHAR(16)
ANNO_ACC	NUMBER(4)
CODFAC_DOC	VARCHAR2(10)
CODICE_CORSO_DI_LAUREA	VARCHAR2(20)
CODICE_INSEGNAMENTO	VARCHAR2(20)
DATA_CONFERMA	DATE
DATA_CHIUSURA	DATE
STATO	VARCHAR2(20) -- (può essere Aperto, Chiuso, Confermato)
ANNOTAZIONI	VARCHAR2(4000)
DATA_CREAZIONE	DATE
DATA_UPDATE	DATE

#### TABELLA ATTIVITA

ID	NOT NULL VARCHAR(64)
ID_REGISTRO	VARCHAR(64)
DATA_ATTIVITA	DATE
DURATA	NUMBER(5,1)
AULA	VARCHAR2(40)
SEDE	VARCHAR2(256)
ARGOMENTO	VARCHAR2(5000)
DATA_CREAZIONE	DATE
DATA_UPDATE	DATE



## Domanda 2

Considerato lo schema E/R e i dati riportati nell'allegato 1, rispondere alle seguenti domande:

1. Descrivere le entità DEPARTMENT e EMPLOYEE, indicando la cardinalità delle relazioni esistenti; definire lo statement DDL per l'aggiunta all'entità EMPLOYEE di un attributo AGE non nullo.
2. Per ogni dipartimento rappresentare i soli dipendenti che lavorano per il dipartimento VENDITE e che beneficiano di un compenso (COMM). La query deve esporre i seguenti attributi:
  - a. Num. del dipartimento, Nome del dipartimento, Location del dipartimento, Nome del dipendente, data assunzione, salario complessivo

## Domanda 3

Descrivere lo script sotto riportato:

```
#!/bin/bash
```

```
su - myUser -c "/mypath/myscript.sh"
cd /mypath/out
FAC=`cat myCollection |grep -oPm2 "(?<=foo)[^<]+" |grep -oP "(?<=toe)[^<]+" | head
-c10` DATE=`date +%Y-%m-%d` if [ $FAC < $DATE ] then
    su - myUser -c "scp -r myCollection
myUser@$remote.node:/remotepath/."
Fi
```

## ALLEGATO COMUNE A TUTTE LE TRACCE

### D A T I - domanda 2

#### DEPARTMENT

DEPTNO	NAME	LOCATION
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

#### EMPLOYEE

EMPNO	NAME	JOB	BOSS	HIREDATE	SALARY	COMM	DEPTNO
8020	BAIT	CLERK	7369	10/04/1980	2700		10
7839	KING	PRESIDENT		17/11/1981	5000		10
7566	JONES	MANAGER	7839	02/04/1981	2975		20
7788	SCOTT	ANALYST	7566	09/12/1982	3000		20
7876	ADAMS	CLERK	7788	12/01/1983	1100		20
7902	FORD	ANALYST	7566	03/12/1981	3000		20
7369	SMITH	CLERK	7902	17/12/1980	800		20
7698	BLAKE	MANAGER	7839	01/05/1981	2850		30
7499	ALLEN	SALESMAN	7698	20/02/1981	1600	300	30
7521	WARD	SALESMAN	7698	22/02/1981	1250	500	30



# UNIVERSITÀ DEGLI STUDI DI MILANO

7654	MARTIN	SALESMAN	7698	28/09/1981	1250	1400	30
7844	TURNER	SALESMAN	7698	08/09/1981	1500	0	30
7900	JAMES	CLERK	7698	03/12/1981	950		30
7782	CLARK	MANAGER	7839	09/06/1981	2450		10
7934	MILLER	CLERK	7782	23/01/1982	1300		10
8000	JOHN	CLERK					
8010	MARY	ANALYST		01/01/2009	2500		

*Get V* *AI* *R*



# UNIVERSITÀ DEGLI STUDI DI MILANO

## D A T I - domanda 2

### PROJECT

PROJECTNO	DESCRIPTION	START_DATE	END_DATE
1001	Development of Novel Magnetic Suspension System	01/01/2006	13/08/2007
1002	Research on thermofluid dynamics in Microdroplets	22/08/2006	20/03/2007
1003	Foundation of Quantum Technology	24/02/2007	31/07/2008
1004	High capacity optical network	01/01/2008	

### PROJECT\_PARTICIPATION

PROJECTNO	EMPNO	START_DATE	END_DATE	ROLE_ID
1001	7902	01/01/2006	30/12/2006	102
1001	7369	01/01/2006	13/08/2007	100
1001	7788	15/05/2006	01/11/2006	100
1002	7876	22/08/2006	20/03/2007	102
1002	7782	22/08/2006	20/03/2007	101
1002	7934	01/01/2007	20/03/2007	101
1003	7566	24/02/2007	31/07/2008	102
1003	7900	24/02/2007	31/01/2007	101
1004	7499	01/01/2008		102
1004	7521	01/05/2008		101
1004	7654	15/04/2008		101
1004	7844	01/02/2008		101
1004	7900	01/03/2008	01/04/2008	101
1004	7900	20/05/2008		101

### SALARYGRADE

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

## D A T I - domanda 2

### ROLE

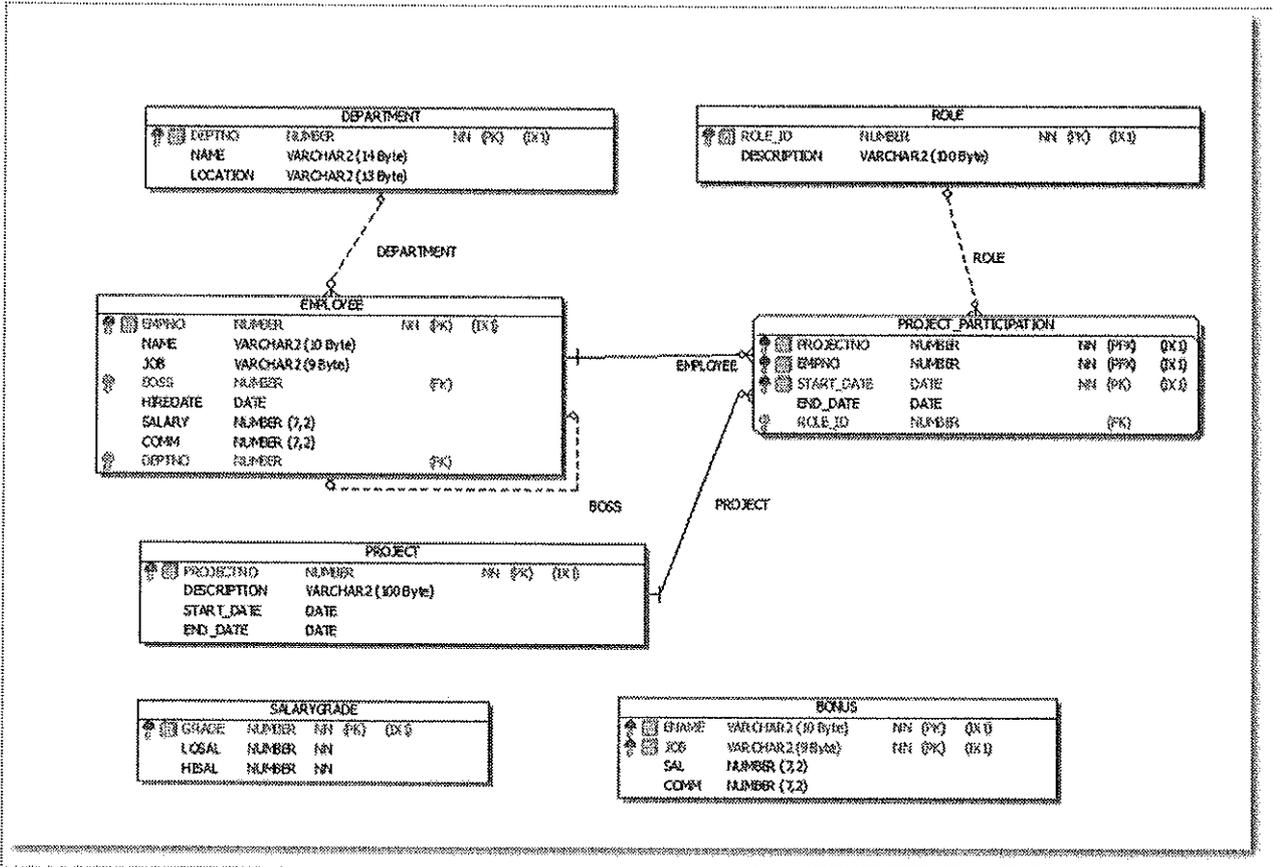
ROLE_ID	DESCRIPTION
100	Developer



# UNIVERSITÀ DEGLI STUDI DI MILANO

101 Researcher

102 Project manager



Schema ER

LA COMMISSIONE

DOTT. SALERNO ANTONIO - PRESIDENTE

DOTT.SSA ARONA ARIANNA - COMPONENTE

DOTT. PUPILLO VINCENZO - COMPONENTE

SIG. FICHERA FRANCESCO - SEGRETARIO

*Antonio Salerno*  
*Arianna Arona*  
*Vincenzo Pupillo*  
*Francesco Fichera*

*ES*