

\*\*\*\*\*ESERCIZIO 1\*\*\*\*\*

```

WITH AUTORE_ARTICOLI_NUMERO (CodAutore, Argomento, Numero) AS
    (SELECT AUTORI_ARTICOLO.CodAutore, ARTICOLO.Argomento, COUNT(*) AS
Numero
    FROM AUTORI_ARTICOLO, ARTICOLO
    WHERE AUTORI_ARTICOLO.CodArticolo=ARTICOLO.CodArticolo
    GROUP BY AUTORI_ARTICOLO.CodAutore, ARTICOLO.Argomento),
AUTORI_NVALIDI (CodAutore) AS
    (SELECT AUTORE_ARTICOLI_NUMERO.CodAutore
    FROM AUTORE_ARTICOLI_NUMERO
    WHERE AUTORE_ARTICOLI_NUMERO.Argomento LIKE 'CyberSecurity' OR
AUTORE_ARTICOLI_NUMERO.Argomento LIKE 'Deep Learning'),
AUTORI_VALIDI (CodAutore) AS
    (SELECT AUTORE_ARTICOLI_NUMERO.CodAutore
    FROM AUTORE_ARTICOLI_NUMERO
    WHERE AUTORE_ARTICOLI_NUMERO.Argomento LIKE 'Data Mining' AND
AUTORE_ARTICOLI_NUMERO.Numero>=4
    AND AUTORE_ARTICOLI_NUMERO.CodAutore NOT IN (SELECT CodAutore
FROM AUTORI_NVALIDI)),
ARTICOLI_PRESENTATI (CodAutore, Conferenza, Edizione,
NumeroPresentazioni) AS
    (SELECT CodAutore, Conferenza, Edizione, COUNT(*) AS
NumeroPresentazioni
    FROM AUTORE_PRESENTA_ARTICOLO
    GROUP BY CodAutore, Conferenza, Edizione)

```

```

SELECT Nome, Cognome, universita, Conferenza, Edizione,
NumeroPresentazioni
FROM AUTORE, AUTORI_VALIDI LEFT OUTER JOIN ARTICOLI_PRESENTATI ON
AUTORI_VALIDI.CodAutore=ARTICOLI_PRESENTATI.CodAutore
WHERE AUTORE.CodAutore = AUTORI_VALIDI.CodAutore

```

\*\*\*\*\*ESERCIZIO 2\*\*\*\*\*

```

WITH
HW_AVG (CodHW, Media) AS
    (SELECT CodHW, AVG(Valutazione)
    FROM VALUTAZIONE_HOMEWORK_CONSEGNATI
    GROUP BY CodHW
    ),
BEST_STUDENTS (MatricolaS, CodHW) AS
    (SELECT VALUTAZIONE_HOMEWORK_CONSEGNATI.MatricolaS,
VALUTAZIONE_HOMEWORK_CONSEGNATI.CodHW
    FROM VALUTAZIONE_HOMEWORK_CONSEGNATI, HW_AVG
    WHERE VALUTAZIONE_HOMEWORK_CONSEGNATI.Valutazione>=HW_AVG.Media AND
VALUTAZIONE_HOMEWORK_CONSEGNATI.CodHW=HW_AVG.CodHW
    )

```

```

SELECT Nome, Cognome, Titolo, Argomento
FROM STUDENTE
JOIN BEST_STUDENTS ON STUDENTE.MatricolaS=BEST_STUDENTS.MatricolaS
JOIN HOMEWORK_DA_CONSEGNARE ON
BEST_STUDENTS.CodHW=HOMEWORK_DA_CONSEGNARE.CodHW

```

\*\*\*\*\*ESERCIZIO 3\*\*\*\*\*

```

WITH HWBASIDIDATI (CodHW, GiorniAnticipo, MatricolaS, CodDocente,
Valutazione) AS (
    SELECT VALUTAZIONE_HOMEWORK_CONSEGNATI.CodHW,
        HOMEWORK_DA_CONSEGNARE.DataScadenzaPrevista -
VALUTAZIONE_HOMEWORK_CONSEGNATI.DataConsegna,
        MatricolaS,
        CodDoc,
        Valutazione
    FROM VALUTAZIONE_HOMEWORK_CONSEGNATI
    JOIN HOMEWORK_DA_CONSEGNARE ON VALUTAZIONE_HOMEWORK_CONSEGNATI.CodHW
= HOMEWORK_DA_CONSEGNARE.CodHW
    WHERE HOMEWORK_DA_CONSEGNARE.Argomento = 'basi di dati'
        AND VALUTAZIONE_HOMEWORK_CONSEGNATI.DataConsegna <
HOMEWORK_DA_CONSEGNARE.DataScadenzaPrevista
),
HWSTAT (Cognome, MediaVal, NDocenti, AVGGiorniAnticipo) AS (
    SELECT Cognome, AVG(Valutazione), COUNT(DISTINCT CodDocente),
AVG(GiorniAnticipo)
    FROM HWBASIDIDATI
    JOIN STUDENTE ON HWBASIDIDATI.MatricolaS = STUDENTE.MatricolaS
    GROUP BY HWBASIDIDATI.MatricolaS, STUDENTE.Cognome
    HAVING COUNT(CodHW) = (
        SELECT COUNT(*)
        FROM HOMEWORK_DA_CONSEGNARE
        WHERE HOMEWORK_DA_CONSEGNARE.Argomento = 'basi di dati'
    )
)

SELECT *
FROM HWSTAT;

```

\*\*\*\*\*ESERCIZIO 4\*\*\*\*\*

```

WITH
ORDER_NUM (Citta, CodR, OrderNum) AS (
    SELECT Citta, RISTORANTE.CodR, COUNT(CodO)
    FROM ORDINI JOIN RISTORANTE ON ORDINI.CodR=RISTORANTE.CodR
    GROUP BY Citta, RISTORANTE.CodR),
MAX_ORDERS_FORCITY (Citta, MaxOrders) AS (
    SELECT Citta, MAX(OrderNum)
    FROM ORDER_NUM
    GROUP BY Citta),
BEST_RESTAURANTS AS (
    SELECT CodR
    FROM ORDER_NUM JOIN MAX_ORDERS_FORCITY
        ON ORDER_NUM.Citta=MAX_ORDERS_FORCITY.Citta
        AND ORDER_NUM.OrderNum=MAX_ORDERS_FORCITY.MaxOrders),
DAILY_CASH_IN (CodR, Giorno, Incasso) AS (
    SELECT ORDINI.CodR, Giorno, SUM(Prezzo)
    FROM BEST_RESTAURANTS JOIN ORDINI ON
BEST_RESTAURANTS.CodR=ORDINI.CodR
    GROUP BY ORDINI.CodR, Giorno)

SELECT NomeR, Giorno, Incasso
FROM DAILY_CASH_IN JOIN RISTORANTE ON DAILY_CASH_IN.CodR=RISTORANTE.CodR
ORDER BY NomeR, Giorno

```

\*\*\*\*\*ESERCIZIO 5\*\*\*\*\*

```
create or replace TRIGGER UPDATE_RIASSUNTO_RECENSIONI
AFTER INSERT ON RECENSIONE_HOTEL
FOR EACH ROW
--NO CONDIZIONI SU WHERE
DECLARE
    I NUMBER;    --BOOOL
    PCH NUMBER;  --PUNTEGGIO COMPLESSIVO HOTEL
    PMC NUMBER;  --PUNTEGGIO MEDIO CATEGORIA
BEGIN
    SELECT COUNT(*) INTO I
    FROM RIASSUNTO_RECENSIONI
    WHERE CODH=:NEW.CODH;

    IF (I=0) THEN
        INSERT INTO RIASSUNTO_RECENSIONI (CodH, NumeroRecensioni,
PunteggioComplessivo)
            VALUES (:NEW.CODH, 1, :NEW.PUNTEGGIO);
    ELSE
        UPDATE RIASSUNTO_RECENSIONI
        SET NumeroRecensioni=NumeroRecensioni+1,
            PunteggioComplessivo=PunteggioComplessivo+:NEW.PUNTEGGIO
        WHERE CodH=:NEW.CodH;
    END IF;

    SELECT PunteggioComplessivo INTO PCH
    FROM RIASSUNTO_RECENSIONI
    WHERE CODH=:NEW.CODH;

    SELECT AVG(RIASSUNTO_RECENSIONI.PunteggioComplessivo) INTO PMC
    FROM HOTEL JOIN RIASSUNTO_RECENSIONI ON
HOTEL.CodH=RIASSUNTO_RECENSIONI.CodH
    GROUP BY HOTEL.Categoria
    HAVING HOTEL.Categoria=(
        SELECT Categoria
        FROM HOTEL
        WHERE HOTEL.CodH=:NEW.CodH
    );

    INSERT INTO NOTIFICA_RECENSIONI_COMPLESSIVE (CodH, DataRecensione,
PunteggioComplessivoHotel, PunteggioMedioCategoria)
        VALUES (:NEW.CODH, :NEW.DATARECENSIONE, PCH, PMC);
END;
/
```