netstat  -ano  |  findstr  < ***Port Number*** >

taskkill  /F  /PID  < ***Process Id*** >

Start Transaction;

INSERT INTO anagrafica

(Nome, Cognome, Email, Cellulare, CodiceFiscale)

VALUES

('Giacomo', 'Salerno', 'giacomo.ermejo@yahoo.it', '+39 3451987443', 'SLNGCMTGHF45DRET');

-- SET @pk\_ID = (SELECT Max(ID\_ANAGRAFICA) FROM anagrafica);

SET @pk\_ID =(SELECT LAST\_INSERT\_ID());

INSERT INTO an\_indirizzi

(p\_Anagrafica, Indirizzo, CAP, Comune, Provincia, Interno, Citofono)

VALUE(@pk\_ID, 'Vigna Pia 88', '00152', 'Portu', 'RM', '3', 'Vice');

-- set @pk\_INDIRIZZO = (SELECT Max(ID\_INDIRIZZO) FROM an\_indirizzi);

set @pk\_INDIRIZZO = (SELECT LAST\_INSERT\_ID());

set @anno = (SELECT YEAR(CURDATE()));

set @NumeroOrdine = (SELECT IFNULL(Max(n\_Ordine),0) + 1 FROM ordini

WHERE YEAR(dt\_Ordine) = @anno);

INSERT INTO ordini

(p\_Indirizzo, n\_Ordine, AnnoOrdine, Sconto)

VALUE

(@pk\_INDIRIZZO, @NumeroOrdine, @anno, 0);

set @Ordine = (SELECT LAST\_INSERT\_ID());

INSERT INTO ordini\_dettagli

(p\_Ordine, p\_Articolo, Prezzo\_Articolo, Quantita)

VALUES

(@Ordine, 3, (SELECT PrezzoArticolo FROM articoli where ID\_ARTICOLO = 3), 1),

(@Ordine, 1, (SELECT PrezzoArticolo FROM articoli where ID\_ARTICOLO = 1), 2),

(@Ordine, 5, (SELECT PrezzoArticolo FROM articoli where ID\_ARTICOLO = 5), 1);

UPDATE articoli SET Giacenza = Giacenza - 1 WHERE ID\_ARTICOLO = 3;

UPDATE articoli SET Giacenza = Giacenza - 2 WHERE ID\_ARTICOLO = 1;

UPDATE articoli SET Giacenza = Giacenza - 1 WHERE ID\_ARTICOLO = 5;

SELECT @Ordine AS PK\_ORDINE;

SELECT CAP, Sum(QuantitaArticolo \* PrezzoArticolo), ID\_ORDINE

FROM anagrafica

left join ordini ON ID\_ANAGRAFICA = p\_Cliente

left join ordini\_dettagli ON ID\_ORDINE = p\_Ordine

WHERE CAP = 100

GROUP BY CAP, ID\_ORDINE;

SELECT CAP, count(DISTINCT ID\_ORDINE), Sum(QuantitaArticolo \* PrezzoArticolo)

FROM anagrafica

left join ordini ON ID\_ANAGRAFICA = p\_Cliente

left join ordini\_dettagli ON ID\_ORDINE = p\_Ordine

WHERE CAP = 100

GROUP BY CAP;

-- DISPONIBILISELECT ID\_LIBRO, titolo, codice, dt\_Prestito, dt\_Riconsegna FROM libri left join copie ON p\_Libro = ID\_LIBRO left join prestiti ON p\_Copia\_pre = ID\_COPIA WHERE (not dt\_prestito IS NULL AND not dt\_Riconsegna is NULL) OR (dt\_prestito is NULL); -- PRESTATISELECT ID\_LIBRO, titolo, codice, dt\_Prestito, dt\_Riconsegna FROM libri left join copie ON p\_Libro = ID\_LIBRO left join prestiti ON p\_Copia\_pre = ID\_COPIA WHERE (not dt\_prestito IS NULL AND dt\_Riconsegna is NULL);

-- PRESTATOSELECT ID\_LIBRO, titolo, ID\_COPIA, codice, dt\_Prestito, dt\_Riconsegna FROM libri left join copie ON p\_Libro = ID\_LIBRO left join prestiti ON p\_Copia\_pre = ID\_COPIA WHERE not dt\_prestito IS NULL AND dt\_Riconsegna is NULL;

-- DISPONIBILISELECT ID\_LIBRO, titolo, ID\_COPIA, codice, dt\_Prestito, dt\_Riconsegna FROM libri left join copie ON p\_Libro = ID\_LIBRO left join prestiti ON p\_Copia\_pre = ID\_COPIA WHERE NOT ID\_COPIA IN ( SELECT ID\_COPIA FROM libri left join copie ON p\_Libro = ID\_LIBRO left join prestiti ON p\_Copia\_pre = ID\_COPIA WHERE not dt\_prestito IS NULL AND dt\_Riconsegna is NULL GROUP BY ID\_COPIA );