**QUERY**

1. OK

CREATE DEFINER=`root`@`localhost` PROCEDURE `inserirePaziente`(codfis varchar(20), nome varchar(12), cognome varchar(12),

indirizzo varchar(45), dataNascita date, numTel varchar(10), sesso varchar(8), eta int)

BEGIN

INSERT INTO paziente (codice\_fiscale, nome, cognome, indirizzo, data\_nascita, numero\_telefonico,

sesso,eta) VALUES (codfis, nome, cognome, indirizzo, dataNascita, numTel, sesso, eta);

END

1. OK

CREATE DEFINER=`root`@`localhost` PROCEDURE `modificaNumEIndirizzoOdontoiatra`(codFis varchar(20), matr varchar(6), nuovoNumero varchar(10),

nuovoIndirizzo varchar(45))

BEGIN

UPDATE odontoiatra

SET indirizzo = nuovoIndirizzo, numero\_telefonico = nuovoNumero

WHERE codice\_fiscale = codFis and matricola\_odontoiatra = matr;

END

1. OK

CREATE DEFINER=`root`@`localhost` PROCEDURE `elencoVisite`(codFisPaziente varchar(20), dataInizioIntervallo date,

dataFineIntervallo date )

BEGIN

SELECT a.codice\_appuntamento, a.tipologia, a.data

FROM appuntamento AS a, paziente AS p

WHERE p.codice\_fiscale = codFisPaziente and a.è\_cancellato = 0 and a.codice\_fiscale\_paziente = p. codice\_fiscale and

(data >= dataInizioIntervallo and data <= dataFineIntervallo)

ORDER BY a.data;

END

1. OK

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `elencovisitemensili` AS

SELECT

`a`.`codice\_fiscale\_odontoiatra` AS `codiceFiscaleOdontoiatra`,

`a`.`codice\_matricola\_odontoiatra` AS `matricolaOdontoiatra`,

MONTHNAME(`a`.`data`) AS `mese`,

COUNT(0) AS `numeroVisite`

FROM

`appuntamento` `a`

WHERE

(`a`.`è\_cancellato` = 0)

GROUP BY `a`.`codice\_fiscale\_odontoiatra` , `a`.`codice\_matricola\_odontoiatra` , MONTHNAME(`a`.`data`)

SELECT \*

FROM elencovisitemensili

1. OK

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `spesamassimasingolavisita` AS

SELECT

`a`.`codice\_fiscale\_paziente` AS `codiceFiscale`,

`p`.`importo` AS `importo`

FROM

(`appuntamento` `a`

JOIN `pagamento` `p`)

WHERE

((`a`.`codice\_appuntamento` = `p`.`codice\_appuntamento\_pagamento`)

AND (`p`.`importo` = (SELECT

MAX(`p1`.`importo`)

FROM

`pagamento` `p1`)))

GROUP BY `a`.`codice\_fiscale\_paziente`

SELECT \*

FROM spesamassimasingolavisita

1. OK

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `pazientevisitatoduevolte` AS

SELECT

`a`.`codice\_fiscale\_odontoiatra` AS `codfis\_odontoiatra`,

`a`.`codice\_fiscale\_paziente` AS `codfis\_paziente`,

WEEK(`a`.`data`, 0) AS `settimana\_visita`

FROM

`appuntamento` `a`

WHERE

((`a`.`è\_cancellato` = 0)

AND `a`.`codice\_fiscale\_odontoiatra` IN (SELECT

`a1`.`codice\_fiscale\_odontoiatra`

FROM

`appuntamento` `a1`

WHERE

((`a1`.`codice\_appuntamento` <> `a`.`codice\_appuntamento`)

AND (`a1`.`è\_cancellato` = 0)

AND (`a1`.`codice\_fiscale\_odontoiatra` = `a`.`codice\_fiscale\_odontoiatra`)

AND (`a1`.`codice\_fiscale\_paziente` = `a`.`codice\_fiscale\_paziente`)

AND (WEEK(`a1`.`data`, 0) = WEEK(`a`.`data`, 0)))))

GROUP BY `a`.`codice\_fiscale\_odontoiatra`

1. OK

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `pazientehaeliminatopiuappuntamenti` AS

SELECT

`a`.`codice\_fiscale\_paziente` AS `codiceFiscalePaziente`,

COUNT(0) AS `numeroAppuntamentiCancellati`

FROM

`appuntamento` `a`

WHERE

(`a`.`è\_cancellato` = 1)

GROUP BY `a`.`codice\_fiscale\_paziente`

HAVING COUNT(0) >= ALL (SELECT

COUNT(0)

FROM

`appuntamento` `a1`

WHERE

(`a1`.`è\_cancellato` = 1)

GROUP BY `a1`.`codice\_fiscale\_paziente`)

SELECT \*

FROM pazientehaeliminatopiuappuntamenti

1. OK

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `pazientenonhamaispostatounappuntamento` AS

SELECT

`a`.`codice\_fiscale\_paziente` AS `codiceFiscalePaziente`

FROM

`appuntamento` `a`

WHERE

((`a`.`è\_cancellato` = 0)

AND (NOT (`a`.`codice\_fiscale\_paziente` IN (SELECT

`a1`.`codice\_fiscale\_paziente`

FROM

((`appuntamento` `a1`

JOIN `appuntamento` `a2`)

JOIN `sostituitoda` `s`)

WHERE

((`a1`.`è\_cancellato` = 0)

AND (`a2`.`è\_cancellato` = 0)

AND (`a1`.`codice\_fiscale\_paziente` = `a`.`codice\_fiscale\_paziente`)

AND (`a2`.`codice\_fiscale\_paziente` = `a`.`codice\_fiscale\_paziente`)

AND (`s`.`codice\_appuntamento\_ha\_sostituito` = `a1`.`codice\_appuntamento`)

AND (`s`.`codice\_appuntamento\_è\_sostituito` = `a2`.`codice\_appuntamento`))))))

GROUP BY `a`.`codice\_fiscale\_paziente`

SELECT \*

FROM pazientenonhamaispostatounappuntamento

1. OK

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `conta\_visite\_per\_cura\_e\_paziente` AS

SELECT

`app`.`codice\_fiscale\_paziente` AS `codicePaziente`,

`s`.`codice\_cura\_stabilita` AS `codiceCura`,

COUNT(0) AS `numeroVisite`

FROM

(`stabilita` `s`

JOIN `appuntamento` `app`)

WHERE

((`app`.`è\_cancellato` = 0)

AND (`s`.`codice\_appuntamento\_per\_cura` = `app`.`codice\_appuntamento`))

GROUP BY `app`.`codice\_fiscale\_paziente`

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `importocomplessivoappuntamento` AS

SELECT

`p`.`codice\_appuntamento\_pagamento` AS `codiceAppuntamento`,

`p`.`importo` AS `importoComplessivo`

FROM

`pagamento` `p`

GROUP BY `p`.`codice\_appuntamento\_pagamento`

CREATE DEFINER=`root`@`localhost` PROCEDURE `numero\_visite\_e\_importo\_cura\_paziente`(codFis VARCHAR(20), codCura VARCHAR(6))

BEGIN

SELECT contVisite.codicePaziente, contVisite.codiceCura, contVisite.numeroVisite, SUM(impCompl.importoComplessivo) AS `importoComplessivoVersato`

FROM conta\_visite\_per\_cura\_e\_paziente AS contVisite, importocomplessivoappuntamento AS impCompl, appuntamento AS app

WHERE app.è\_cancellato = 0 AND contVisite.codicePaziente = codFis AND contVisite.codiceCura = codCura AND

contVisite.codicePaziente = app.codice\_fiscale\_paziente AND

impCompl.codiceAppuntamento = app.codice\_appuntamento

GROUP BY contVisite.codicePaziente, contVisite.codiceCura;

END

CALL numero\_visite\_e\_importo\_cura\_paziente("SSSLNE71A46H558D","423153");

10)

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `appuntamenti\_preliminari\_ultimo\_anno` AS

SELECT

`app`.`codice\_fiscale\_paziente` AS `codiceFiscalePaziente`,

`app`.`codice\_appuntamento` AS `codiceAppuntamentoUltimoAnno`,

`app`.`tipologia` AS `tipologiaAppuntamento`

FROM

`appuntamento` `app`

WHERE

((`app`.`è\_cancellato` = 0)

AND (`app`.`tipologia` = 'Preliminare')

AND (YEAR(`app`.`data`) = YEAR(CURDATE()))

AND (NOT (`app`.`codice\_fiscale\_paziente` IN (SELECT

`appunt`.`codice\_fiscale\_paziente`

FROM

`appuntamento` `appunt`

WHERE

((`appunt`.`è\_cancellato` = 0)

AND (`appunt`.`codice\_fiscale\_paziente` = `app`.`codice\_fiscale\_paziente`)

AND (`appunt`.`tipologia` <> 'Preliminare'))))))

GROUP BY `app`.`codice\_fiscale\_paziente` , `app`.`codice\_appuntamento`

CREATE DEFINER=`root`@`localhost` PROCEDURE `elimina\_appuntamenti\_preliminari`()

BEGIN

DELETE

FROM appuntamento

WHERE è\_cancellato = 0 AND codice\_appuntamento IN (

SELECT appPre.codiceAppuntamentoUltimoAnno

FROM appuntamenti\_preliminari\_ultimo\_anno AS appPre

);

END

USE `studioodontoiatrico`;

SET GLOBAL event\_scheduler = ON;

DELIMITER $$

CREATE EVENT elimina\_appuntamenti ON SCHEDULE EVERY 365 DAY

DO

BEGIN

CALL elimina\_appuntamenti\_preliminari();

END$$

DELIMITER ;

11)

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `appuntamento\_e\_relativo\_intervento\_con\_costo\_intervento` AS

SELECT

`rich`.`codice\_appuntamento\_per\_intervento` AS `codiceAppuntamentoPerIntervento`,

`rich`.`codice\_intervento\_richiesto` AS `codiceIntervento`,

`interv`.`costo\_effettivo\_intervento` AS `costoEffettivoIntervento`

FROM

(`richiede` `rich`

JOIN `intervento` `interv`)

WHERE

(`rich`.`codice\_intervento\_richiesto` = `interv`.`codice\_intervento`)

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `importo\_pagato\_per\_intervento` AS

SELECT

`appeintervento`.`codiceIntervento` AS `codiceInterventoPagato`,

SUM(`pag`.`importo`) AS `ImportoPagatoPerIntervento`

FROM

((`appuntamento` `app`

JOIN `appuntamento\_e\_relativo\_intervento\_con\_costo\_intervento` `appeintervento`)

JOIN `pagamento` `pag`)

WHERE

((`app`.`è\_cancellato` = 0)

AND (`app`.`codice\_appuntamento` = `appeintervento`.`codiceAppuntamentoPerIntervento`)

AND (`pag`.`codice\_appuntamento\_pagamento` = `app`.`codice\_appuntamento`))

GROUP BY `appeintervento`.`codiceIntervento`

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `numero\_interventi\_pagati` AS

SELECT

COUNT(0) AS `numero\_interventi\_saldati`

FROM

`importo\_pagato\_per\_intervento` `impopag`

WHERE

`impopag`.`codiceInterventoPagato` IN (SELECT

`appecostointer`.`codiceIntervento`

FROM

`appuntamento\_e\_relativo\_intervento\_con\_costo\_intervento` `appecostointer`

WHERE

(`impopag`.`ImportoPagatoPerIntervento` >= `appecostointer`.`costoEffettivoIntervento`))

CREATE DEFINER=`root`@`localhost` FUNCTION `percentuale\_interventi\_non\_saldati`(numeroInterventi INTEGER, numeroInterventiPagati INTEGER) RETURNS int(11)

BEGIN

DECLARE percentuale FLOAT;

SET percentuale = (numeroInterventiPagati/numeroInterventi) \* 100;

RETURN percentuale;

END

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `numero\_interventi\_complessivo` AS

SELECT

COUNT(0) AS `numeroInterventiComplessivo`

FROM

`appuntamento` `app`

WHERE

((`app`.`è\_cancellato` = 0)

AND (`app`.`tipologia` = 'Intervento'))

CREATE DEFINER=`root`@`localhost` PROCEDURE `calcolo\_numero\_interventi\_e\_percentuale\_non\_saldati`()

BEGIN

SELECT intCompl.numeroInterventiComplessivo, percentuale\_interventi\_non\_saldati(intCompl.numeroInterventiComplessivo,

intPag.numeroInterventiPagati) AS 'percentualeInterventiNonSaldati'

FROM numero\_interventi\_complessivo AS intCompl, numero\_interventi\_pagati AS intPag;

END

USE `studioodontoiatrico`;

SET GLOBAL event\_scheduler = ON;

DELIMITER $$

CREATE EVENT mostra\_interventi\_complessivi\_e\_percentuale ON SCHEDULE

EVERY 28 DAY

DO

BEGIN

CALL calcolo\_numero\_interventi\_e\_percentuale\_non\_saldati();

END$$

DELIMITER ;