Example 4.1:

CREATE USER 'BOOKSQL'@'localhost' IDENTIFIED BY 'BOOKSQLPW'

Example 4.2:

GRANT ALL PRIVILEGES

ON \*.\*

TO 'BOOKSQL'@'localhost'

WITH GRANT OPTION

Example 4.3:

CREATE DATABASE TENNIS

Example 4.4:

USE TENNIS

Example 4.5:

CREATE TABLE PLAYERS

(PLAYERNO INTEGER NOT NULL,

NAME CHAR(15) NOT NULL,

INITIALS CHAR(3) NOT NULL,

BIRTH\_DATE DATE ,

SEX CHAR(1) NOT NULL,

JOINED SMALLINT NOT NULL,

STREET VARCHAR(30) NOT NULL,

HOUSENO CHAR(4) ,

POSTCODE CHAR(6) ,

TOWN VARCHAR(30) NOT NULL,

PHONENO CHAR(13) ,

LEAGUENO CHAR(4) ,

PRIMARY KEY (PLAYERNO) )

;

CREATE TABLE TEAMS

(TEAMNO INTEGER NOT NULL,

PLAYERNO INTEGER NOT NULL,

DIVISION CHAR(6) NOT NULL,

PRIMARY KEY (TEAMNO) )

;

CREATE TABLE MATCHES

(MATCHNO INTEGER NOT NULL,

TEAMNO INTEGER NOT NULL,

PLAYERNO INTEGER NOT NULL,

WON SMALLINT NOT NULL,

LOST SMALLINT NOT NULL,

PRIMARY KEY (MATCHNO) )

;

CREATE TABLE PENALTIES

(PAYMENTNO INTEGER NOT NULL,

PLAYERNO INTEGER NOT NULL,

PAYMENT\_DATE DATE NOT NULL,

AMOUNT DECIMAL(7,2) NOT NULL,

PRIMARY KEY (PAYMENTNO) )

;

CREATE TABLE COMMITTEE\_MEMBERS

(PLAYERNO INTEGER NOT NULL,

BEGIN\_DATE DATE NOT NULL,

END\_DATE DATE ,

POSITION CHAR(20) ,

PRIMARY KEY (PLAYERNO, BEGIN\_DATE))

Example 4.6:

INSERT INTO PLAYERS VALUES (

2, 'Everett', 'R', '1948-09-01', 'M', 1975, 'Stoney Road',

'43', '3575NH', 'Stratford', '070-237893', '2411')

;

INSERT INTO PLAYERS VALUES (

6, 'Parmenter', 'R', '1964-06-25', 'M', 1977, 'Haseltine Lane',

'80', '1234KK', 'Stratford', '070-476537', '8467')

;

INSERT INTO PLAYERS VALUES (

7, 'Wise', 'GWS', '1963-05-11', 'M', 1981, 'Edgecombe Way',

'39', '9758VB', 'Stratford', '070-347689', NULL)

;

INSERT INTO PLAYERS VALUES (

8, 'Newcastle', 'B', '1962-07-08', 'F', 1980, 'Station Road',

'4', '6584WO', 'Inglewood', '070-458458', '2983')

;

INSERT INTO PLAYERS VALUES (

27, 'Collins', 'DD', '1964-12-28', 'F', 1983, 'Long Drive',

'804', '8457DK', 'Eltham', '079-234857', '2513')

;

INSERT INTO PLAYERS VALUES (

28, 'Collins', 'C', '1963-06-22', 'F', 1983, 'Old Main Road',

'10', '1294QK', 'Midhurst', '010-659599', NULL)

;

INSERT INTO PLAYERS VALUES (

39, 'Bishop', 'D', '1956-10-29', 'M', 1980, 'Eaton Square',

'78', '9629CD', 'Stratford', '070-393435', NULL)

;

INSERT INTO PLAYERS VALUES (

44, 'Baker', 'E', '1963-01-09', 'M', 1980, 'Lewis Street',

'23', '4444LJ', 'Inglewood', '070-368753', '1124')

;

INSERT INTO PLAYERS VALUES (

57, 'Brown', 'M', '1971-08-17', 'M', 1985, 'Edgecombe Way',

'16', '4377CB', 'Stratford', '070-473458', '6409')

;

INSERT INTO PLAYERS VALUES (

83, 'Hope', 'PK', '1956-11-11', 'M', 1982, 'Magdalene Road',

'16A', '1812UP', 'Stratford', '070-353548', '1608')

;

INSERT INTO PLAYERS VALUES (

95, 'Miller', 'P', '1963-05-14', 'M', 1972, 'High Street',

'33A', '5746OP', 'Douglas', '070-867564', NULL)

;

INSERT INTO PLAYERS VALUES (

100, 'Parmenter', 'P', '1963-02-28', 'M', 1979, 'Haseltine Lane',

'80', '6494SG', 'Stratford', '070-494593', '6524')

;

INSERT INTO PLAYERS VALUES (

104, 'Moorman', 'D', '1970-05-10', 'F', 1984, 'Stout Street',

'65', '9437AO', 'Eltham', '079-987571', '7060')

;

INSERT INTO PLAYERS VALUES (

112, 'Bailey', 'IP', '1963-10-01', 'F', 1984, 'Vixen Road',

'8', '6392LK', 'Plymouth', '010-548745', '1319')

;

INSERT INTO TEAMS VALUES (1, 6, 'first')

;

INSERT INTO TEAMS VALUES (2, 27, 'second')

;

INSERT INTO MATCHES VALUES ( 1, 1, 6, 3, 1)

;

INSERT INTO MATCHES VALUES ( 2, 1, 6, 2, 3)

;

INSERT INTO MATCHES VALUES ( 3, 1, 6, 3, 0)

;

INSERT INTO MATCHES VALUES ( 4, 1, 44, 3, 2)

;

INSERT INTO MATCHES VALUES ( 5, 1, 83, 0, 3)

;

INSERT INTO MATCHES VALUES ( 6, 1, 2, 1, 3)

;

INSERT INTO MATCHES VALUES ( 7, 1, 57, 3, 0)

;

INSERT INTO MATCHES VALUES ( 8, 1, 8, 0, 3)

;

INSERT INTO MATCHES VALUES ( 9, 2, 27, 3, 2)

;

INSERT INTO MATCHES VALUES (10, 2, 104, 3, 2)

;

INSERT INTO MATCHES VALUES (11, 2, 112, 2, 3)

;

INSERT INTO MATCHES VALUES (12, 2, 112, 1, 3)

;

INSERT INTO MATCHES VALUES (13, 2, 8, 0, 3)

;

INSERT INTO PENALTIES VALUES (1, 6, '1980-12-08',100)

;

INSERT INTO PENALTIES VALUES (2, 44, '1981-05-05', 75)

;

INSERT INTO PENALTIES VALUES (3, 27, '1983-09-10',100)

;

INSERT INTO PENALTIES VALUES (4,104, '1984-12-08', 50)

;

INSERT INTO PENALTIES VALUES (5, 44, '1980-12-08', 25)

;

INSERT INTO PENALTIES VALUES (6, 8, '1980-12-08', 25)

;

INSERT INTO PENALTIES VALUES (7, 44, '1982-12-30', 30)

;

INSERT INTO PENALTIES VALUES (8, 27, '1984-11-12', 75)

;

INSERT INTO COMMITTEE\_MEMBERS VALUES ( 6, '1990-01-01', '1990-12-31', 'Secretary')

;

INSERT INTO COMMITTEE\_MEMBERS VALUES ( 6, '1991-01-01', '1992-12-31', 'Member')

;

INSERT INTO COMMITTEE\_MEMBERS VALUES ( 6, '1992-01-01', '1993-12-31', 'Treasurer')

;

INSERT INTO COMMITTEE\_MEMBERS VALUES ( 6, '1993-01-01', NULL, 'Chairman')

;

INSERT INTO COMMITTEE\_MEMBERS VALUES ( 2, '1990-01-01', '1992-12-31', 'Chairman')

;

INSERT INTO COMMITTEE\_MEMBERS VALUES ( 2, '1994-01-01', NULL, 'Member')

;

INSERT INTO COMMITTEE\_MEMBERS VALUES (112, '1992-01-01', '1992-12-31', 'Member')

;

INSERT INTO COMMITTEE\_MEMBERS VALUES (112, '1994-01-01', NULL, 'Secretary')

;

INSERT INTO COMMITTEE\_MEMBERS VALUES ( 8, '1990-01-01', '1990-12-31', 'Treasurer')

;

INSERT INTO COMMITTEE\_MEMBERS VALUES ( 8, '1991-01-01', '1991-12-31', 'Secretary')

;

INSERT INTO COMMITTEE\_MEMBERS VALUES ( 8, '1993-01-01', '1993-12-31', 'Member')

;

INSERT INTO COMMITTEE\_MEMBERS VALUES ( 8, '1994-01-01', NULL, 'Member')

;

INSERT INTO COMMITTEE\_MEMBERS VALUES ( 57, '1992-01-01', '1992-12-31', 'Secretary')

;

INSERT INTO COMMITTEE\_MEMBERS VALUES ( 27, '1990-01-01', '1990-12-31', 'Member')

;

INSERT INTO COMMITTEE\_MEMBERS VALUES ( 27, '1991-01-01', '1991-12-31', 'Treasurer')

;

INSERT INTO COMMITTEE\_MEMBERS VALUES ( 27, '1993-01-01', '1993-12-31', 'Treasurer')

;

INSERT INTO COMMITTEE\_MEMBERS VALUES ( 95, '1994-01-01', NULL, 'Treasurer')

Example 4.7:

SELECT PLAYERNO, NAME, BIRTH\_DATE

FROM PLAYERS

WHERE TOWN = 'Stratford'

ORDER BY NAME

Example 4.8:

SELECT PLAYERNO

FROM PLAYERS

WHERE JOINED > 1980

AND TOWN = 'Stratford'

ORDER BY PLAYERNO

Example 4.9:

SELECT \*

FROM PENALTIES

Example 4.10:

SELECT 33 \* 121

Example 4.11:

UPDATE PENALTIES

SET AMOUNT = 200

WHERE PLAYERNO = 44

;

SELECT PLAYERNO, AMOUNT

FROM PENALTIES

WHERE PLAYERNO = 44

Example 4.12:

DELETE

FROM PENALTIES

WHERE AMOUNT > 100

Example 4.13:

CREATE INDEX PENALTIES\_AMOUNT ON

PENALTIES (AMOUNT)

Example 4.14:

CREATE VIEW NUMBER\_SETS (MATCHNO, DIFFERENCE) AS

SELECT MATCHNO, ABS(WON - LOST)

FROM MATCHES

;

SELECT \*

FROM NUMBER\_SETS

Example 4.15:

GRANT SELECT

ON PLAYERS

TO DIANE

;

GRANT SELECT, UPDATE

ON PLAYERS

TO PAUL

;

GRANT SELECT, UPDATE

ON TEAMS

TO PAUL

;

SELECT \*

FROM TEAMS

Example 4.16:

DROP TABLE MATCHES

Example 4.17:

DROP VIEW NUMBER\_SETS

Example 4.18:

DROP INDEX PENALTIES\_AMOUNT

Example 4.19:

DROP DATABASE TENNIS

Example 4.20:

SELECT @@VERSION

Example 4.21:

SET @@SQL\_MODE = 'PIPES\_AS\_CONCAT'

Example 4.22:

SET @@SQL\_MODE = CONCAT(@@SQL\_MODE,

CASE @@SQL\_MODE WHEN '' THEN '' ELSE ',' END,

'NO\_ZERO\_IN\_DATE')

Example 4.23:

CREATE OR REPLACE VIEW USERS

(USER\_NAME) AS

SELECT DISTINCT UPPER(CONCAT('''',USER,'''@''',HOST,''''))

FROM MYSQL.USER

;

CREATE OR REPLACE VIEW TABLES

(TABLE\_CREATOR, TABLE\_NAME,

CREATE\_TIMESTAMP, COMMENT) AS

SELECT UPPER(TABLE\_SCHEMA), UPPER(TABLE\_NAME),

CREATE\_TIME, TABLE\_COMMENT

FROM INFORMATION\_SCHEMA.TABLES

WHERE TABLE\_TYPE IN ('BASE TABLE','TEMPORARY')

;

CREATE OR REPLACE VIEW COLUMNS

(TABLE\_CREATOR, TABLE\_NAME, COLUMN\_NAME,

COLUMN\_NO, DATA\_TYPE, CHAR\_LENGTH,

`PRECISION`, SCALE, NULLABLE, COMMENT) AS

SELECT UPPER(TABLE\_SCHEMA), UPPER(TABLE\_NAME),

UPPER(COLUMN\_NAME), ORDINAL\_POSITION,

UPPER(DATA\_TYPE), CHARACTER\_MAXIMUM\_LENGTH,

NUMERIC\_PRECISION, NUMERIC\_SCALE, IS\_NULLABLE,

COLUMN\_COMMENT

FROM INFORMATION\_SCHEMA.COLUMNS

;

CREATE OR REPLACE VIEW VIEWS

(VIEW\_CREATOR, VIEW\_NAME, CREATE\_TIMESTAMP,

WITHCHECKOPT, IS\_UPDATABLE, VIEWFORMULA, COMMENT) AS

SELECT UPPER(V.TABLE\_SCHEMA), UPPER(V.TABLE\_NAME),

T.CREATE\_TIME,

CASE

WHEN V.CHECK\_OPTION = 'None' THEN 'NO'

WHEN V.CHECK\_OPTION = 'Cascaded' THEN 'CASCADED'

WHEN V.CHECK\_OPTION = 'Local' THEN 'LOCAL'

ELSE 'Yes'

END, V.IS\_UPDATABLE, V.VIEW\_DEFINITION, T.TABLE\_COMMENT

FROM INFORMATION\_SCHEMA.VIEWS AS V,

INFORMATION\_SCHEMA.TABLES AS T

WHERE V.TABLE\_NAME = T.TABLE\_NAME

AND V.TABLE\_SCHEMA = T.TABLE\_SCHEMA

;

CREATE OR REPLACE VIEW INDEXES

(INDEX\_CREATOR, INDEX\_NAME, CREATE\_TIMESTAMP,

TABLE\_CREATOR, TABLE\_NAME, UNIQUE\_ID, INDEX\_TYPE) AS

SELECT DISTINCT UPPER(I.INDEX\_SCHEMA), UPPER(I.INDEX\_NAME),

T.CREATE\_TIME, UPPER(I.TABLE\_SCHEMA),

UPPER(I.TABLE\_NAME),

CASE

WHEN I.NON\_UNIQUE = 0 THEN 'YES'

ELSE 'NO'

END,

I.INDEX\_TYPE

FROM INFORMATION\_SCHEMA.STATISTICS AS I,

INFORMATION\_SCHEMA.TABLES AS T

WHERE I.TABLE\_NAME = T.TABLE\_NAME

AND I.TABLE\_SCHEMA = T.TABLE\_SCHEMA

;

CREATE OR REPLACE VIEW COLUMNS\_IN\_INDEX

(INDEX\_CREATOR, INDEX\_NAME,

TABLE\_CREATOR, TABLE\_NAME, COLUMN\_NAME,

COLUMN\_SEQ, ORDERING) AS

SELECT UPPER(INDEX\_SCHEMA), UPPER(INDEX\_NAME),

UPPER(TABLE\_SCHEMA), UPPER(TABLE\_NAME),

UPPER(COLUMN\_NAME), SEQ\_IN\_INDEX,

CASE

WHEN COLLATION = 'A' THEN 'ASCENDING'

WHEN COLLATION = 'D' THEN 'DESCENDING'

ELSE 'OTHER'

END

FROM INFORMATION\_SCHEMA.STATISTICS

;

CREATE OR REPLACE VIEW USER\_AUTHS

(GRANTOR, GRANTEE, PRIVILEGE, WITHGRANTOPT) AS

SELECT 'UNKNOWN', UPPER(GRANTEE), PRIVILEGE\_TYPE, IS\_GRANTABLE

FROM INFORMATION\_SCHEMA.USER\_PRIVILEGES

;

CREATE OR REPLACE VIEW DATABASE\_AUTHS

(GRANTOR, GRANTEE, DATABASE\_NAME, PRIVILEGE,

WITHGRANTOPT) AS

SELECT 'UNKNOWN', UPPER(GRANTEE), UPPER(TABLE\_SCHEMA),

PRIVILEGE\_TYPE, IS\_GRANTABLE

FROM INFORMATION\_SCHEMA.SCHEMA\_PRIVILEGES

;

CREATE OR REPLACE VIEW TABLE\_AUTHS

(GRANTOR, GRANTEE, TABLE\_CREATOR, TABLE\_NAME,

PRIVILEGE, WITHGRANTOPT) AS

SELECT 'UNKNOWN', UPPER(GRANTEE), UPPER(TABLE\_SCHEMA),

UPPER(TABLE\_NAME), PRIVILEGE\_TYPE, IS\_GRANTABLE

FROM INFORMATION\_SCHEMA.TABLE\_PRIVILEGES

;

CREATE OR REPLACE VIEW COLUMN\_AUTHS

(GRANTOR, GRANTEE, TABLE\_CREATOR, TABLE\_NAME,

COLUMN\_NAME, PRIVILEGE, WITHGRANTOPT) AS

SELECT 'UNKNOWN', UPPER(GRANTEE), UPPER(TABLE\_SCHEMA),

UPPER(TABLE\_NAME), UPPER(COLUMN\_NAME),

PRIVILEGE\_TYPE, IS\_GRANTABLE

FROM INFORMATION\_SCHEMA.COLUMN\_PRIVILEGES

Example 4.24:

SELECT COLUMN\_NAME, DATA\_TYPE, COLUMN\_NO

FROM COLUMNS

WHERE TABLE\_NAME = 'PLAYERS'

AND TABLE\_CREATOR = 'TENNIS'

ORDER BY COLUMN\_NO

Example 4.25:

SELECT INDEX\_NAME

FROM INDEXES

WHERE TABLE\_NAME = 'PENALTIES'

AND TABLE\_CREATOR = 'TENNIS'

Example 4.26:

USE INFORMATION\_SCHEMA

;

SELECT DISTINCT INDEX\_NAME

FROM STATISTICS

WHERE TABLE\_NAME = 'PENALTIES'

Example 4.27:

SELECT TABLE\_NAME

FROM TABLES

WHERE TABLE\_SCHEMA = 'INFORMATION\_SCHEMA'

ORDER BY TABLE\_NAME

Example 4.28:

SHOW COLUMNS FROM PLAYERS

Example 4.29:

SHOW INDEX FROM PENALTIES

;

SHOW DATABASES

;

SHOW TABLES

;

SHOW CREATE TABLE PLAYERS

;

SHOW INDEX FROM PLAYERS

;

SHOW GRANTS FOR BOOKSQL@localhost

;

SHOW PRIVILEGES

Example 4.30:

SELECT 10 / 0

Example 5.1:

CREATE TABLE INCORRECT\_DATES (COLUMN1 DATE)

;

INSERT INTO INCORRECT\_DATES VALUES ('2004-13-12')

;

SELECT COLUMN1

FROM INCORRECT\_DATES

;

CREATE TABLE TIME\_TABLE (COLUMN1 TIME)

;

INSERT INTO TIME\_TABLE VALUES ('23:59:59.5912')

;

SELECT COLUMN1 FROM TIME\_TABLE

Example 5.2:

CREATE TABLE TZ (COL1 TIMESTAMP)

;

INSERT INTO TZ VALUES ('2005-01-01 12:00:00')

;

SELECT \* FROM TZ

;

SET @@TIME\_ZONE = '+10:00'

;

SELECT \* FROM TZ

Example 5.3:

SELECT @@TIME\_ZONE

Example 5.4:

SELECT TRUE, FALSE

Example 5.5:

SELECT MATCHNO, WON - LOST

FROM MATCHES

WHERE WON = LOST + 2

Example 5.6:

SELECT TEAMNO, DIVISION

FROM TEAMS

Example 5.7:

SELECT TEAMNO AS TEAM\_NUMBER, DIVISION AS DIVISION\_OF\_TEAM

FROM TEAMS

Example 5.8:

SELECT PAYMENTNO, AMOUNT \* 100 AS CENTS

FROM PENALTIES

Example 5.9:

SELECT MATCHNO AS PRIMKEY,

80 AS EIGHTY,

WON - LOST AS DIFFERENCE,

TIME('23:59:59') AS ALMOST\_MIDNIGHT,

'TEXT' AS TEXT

FROM MATCHES

WHERE MATCHNO <= 4

Example 5.10:

SELECT PAYMENTNO, AMOUNT \* 100 AS CENTS

FROM PENALTIES

GROUP BY CENTS

ORDER BY CENTS

Example 5.11:

SET @PLAYERNO = 7

Example 5.12:

SELECT NAME, TOWN, POSTCODE

FROM PLAYERS

WHERE PLAYERNO < @PLAYERNO

Example 5.13:

SELECT @PLAYERNO

Example 5.14:

SET @@GLOBAL.SQL\_WARNINGS = TRUE

Example 5.15:

SET @@SESSION.DATETIME\_FORMAT = '%D-%m-%Y %H:%i:%s'

;

SELECT @@SESSION.DATETIME\_FORMAT

;

SELECT @@GLOBAL.DATETIME\_FORMAT

Example 5.16:

SET @@SESSION.DATETIME\_FORMAT = DEFAULT

Example 5.17:

SELECT \*

FROM USER\_AUTHS

WHERE GRANTEE = CURRENT\_USER

Example 5.18:

SELECT CURRENT\_USER

Example 5.19:

SELECT \*

FROM PENALTIES

WHERE PAYMENT\_DATE = CURRENT\_DATE

Example 5.20:

SELECT PLAYERNO,

CASE SEX

WHEN 'F' THEN 'Female'

ELSE 'Male' END AS SEX,

NAME

FROM PLAYERS

WHERE JOINED > 1980

;

SELECT PLAYERNO,

CASE SEX

WHEN 'F' THEN 'Female' END AS FEMALES,

NAME

FROM PLAYERS

WHERE JOINED > 1980

Example 5.21:

SELECT PLAYERNO, TOWN, BIRTH\_DATE,

CASE TOWN

WHEN 'Stratford' THEN 0

WHEN 'Plymouth' THEN 1

WHEN 'Inglewood' THEN 2

ELSE 3

END AS P,

CASE TOWN

WHEN 'Stratford' THEN

CASE BIRTH\_DATE

WHEN '1948-09-01' THEN 'Old Stratforder'

ELSE 'Young Stratforder' END

WHEN 'Inglewood' THEN

CASE BIRTH\_DATE

WHEN '1962-07-08' THEN 'Old Inglewooder'

ELSE 'Young Inglewooder' END

ELSE 'Rest'

END AS TYPE

FROM PLAYERS

Example 5.22:

SELECT PLAYERNO, JOINED,

CASE

WHEN JOINED < 1980 THEN 'Seniors'

WHEN JOINED < 1983 THEN 'Juniors'

ELSE 'Children' END AS AGE\_GROUP

FROM PLAYERS

ORDER BY JOINED

Example 5.23:

SELECT PLAYERNO, JOINED, TOWN,

CASE

WHEN JOINED >= 1980 AND JOINED <= 1982

THEN 'Seniors'

WHEN TOWN = 'Eltham'

THEN 'Elthammers'

WHEN PLAYERNO < 10

THEN 'First members'

ELSE 'Rest' END

FROM PLAYERS

Example 5.24:

SELECT (PLAYERNO), (((NAME)))

FROM PLAYERS

Example 5.25:

SELECT PAYMENTNO, YEAR(PAYMENT\_DATE)

FROM PENALTIES

WHERE YEAR(PAYMENT\_DATE) > 1980

Example 5.26:

SELECT PLAYERNO, CONCAT(LEFT(INITIALS, 1), '. ', NAME)

AS FULL\_NAME

FROM PLAYERS

WHERE LEFT(NAME, 1) = 'B'

Example 5.27:

SELECT INITIALS, NAME, COALESCE(LEAGUENO, 1)

FROM PLAYERS

WHERE Town = 'Stratford'

Example 5.28:

SELECT PLAYERNO, DAYNAME(BIRTH\_DATE),

MONTHNAME(BIRTH\_DATE), DAYOFYEAR(BIRTH\_DATE)

FROM PLAYERS

WHERE PLAYERNO < 10

Example 5.29:

SELECT PLAYERNO, BIRTH\_DATE,

ADDDATE(BIRTH\_DATE, INTERVAL 7 DAY)

FROM PLAYERS

WHERE DAYNAME(BIRTH\_DATE) = 'Saturday'

Example 5.30:

SELECT PLAYERNO, BEGIN\_DATE, END\_DATE,

DATEDIFF(END\_DATE, BEGIN\_DATE)

FROM COMMITTEE\_MEMBERS

WHERE DATEDIFF(END\_DATE, BEGIN\_DATE) > 500

OR (END\_DATE IS NULL AND

DATEDIFF(CURRENT\_DATE, BEGIN\_DATE) > 500)

ORDER BY PLAYERNO

;

SELECT PLAYERNO, BEGIN\_DATE, END\_DATE,

DATEDIFF(COALESCE(END\_DATE, CURRENT\_DATE),

BEGIN\_DATE)

FROM COMMITTEE\_MEMBERS

WHERE DATEDIFF(COALESCE(END\_DATE, CURRENT\_DATE),

BEGIN\_DATE)

> 500

ORDER BY PLAYERNO

Example 5.31:

SELECT PAYMENTNO

FROM PENALTIES

WHERE AMOUNT > 50

Example 5.32:

SELECT CONCAT(RTRIM(NAME), CAST(BIRTH\_DATE AS CHAR(10)))

FROM PLAYERS

WHERE TOWN = 'Inglewood'

Example 5.33:

UPDATE PLAYERS

SET LEAGUENO = NULL

WHERE PLAYERNO = 2

Example 5.34:

SELECT TEAMNO, CAST(NULL AS CHAR)

FROM TEAMS

Example 5.35:

SELECT MATCHNO, WON, LOST

FROM MATCHES

WHERE WON >= LOST \* 2

Example 5.36:

SELECT 50 << 2

Example 5.37:

SELECT B'11' << 3

Example 5.38:

SELECT CONV(6,10,2), CONV(10,10,2), BIN(8), BIN(10)

Example 5.39:

SELECT CONV(1001,2,10), CONV(111,2,10)

Example 5.40:

SELECT PLAYERNO

FROM PLAYERS

WHERE PLAYERNO & 1

Example 5.41:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE PLAYERNO = (PLAYERNO >> 1) << 1

Example 5.42:

SELECT MATCHNO, TEAMNO, MATCHNO | TEAMNO,

MATCHNO & TEAMNO, MATCHNO ^ TEAMNO

FROM MATCHES

Example 5.43:

SELECT PLAYERNO, TOWN || ' ' || STREET || ' ' || HOUSENO

FROM PLAYERS

WHERE TOWN = 'Stratford'

Example 5.44:

SELECT PAYMENTNO, PAYMENT\_DATE, PAYMENT\_DATE + INTERVAL 7 DAY

FROM PENALTIES

WHERE PAYMENTNO > 5

Example 5.45:

SELECT PAYMENTNO, PAYMENT\_DATE

FROM PENALTIES

WHERE PAYMENT\_DATE >= '1982-12-25'

AND PAYMENT\_DATE <= '1982-12-25' + INTERVAL 6 DAY

Example 5.46:

SELECT '2004-13-12' + INTERVAL 1 DAY

;

SHOW WARNINGS

Example 5.47:

CREATE TABLE MATCHES\_SPECIAL

(MATCHNO INTEGER NOT NULL,

TEAMNO INTEGER NOT NULL,

PLAYERNO INTEGER NOT NULL,

WON SMALLINT NOT NULL,

LOST SMALLINT NOT NULL,

START\_DATE DATE NOT NULL,

START\_TIME TIME NOT NULL,

END\_TIME TIME NOT NULL,

PRIMARY KEY (MATCHNO))

;

INSERT INTO MATCHES\_SPECIAL VALUES

(1, 1, 6, 3, 1, '2004-10-25', '14:10:12', '16:50:09')

;

INSERT INTO MATCHES\_SPECIAL VALUES

(2, 1, 44, 3, 2, '2004-10-25', '17:00:00', '17:55:48')

Example 5.48:

SELECT MATCHNO, START\_TIME,

ADDTIME(START\_TIME, '08:00:00')

FROM MATCHES\_SPECIAL

Example 5.49:

SELECT MATCHNO, END\_TIME

FROM MATCHES\_SPECIAL

WHERE ADDTIME(END\_TIME, '06:30:00') <= '24:00:00'

Example 5.50:

CREATE TABLE TSTAMP (COL1 TIMESTAMP)

;

SET @TIME = TIMESTAMP('1980-12-08 23:59:59.59')

;

INSERT INTO TSTAMP VALUES (@TIME + INTERVAL 3 MICROSECOND)

;

SELECT COL1, COL1 + INTERVAL 3 MICROSECOND FROM TSTAMP

Example 5.51:

SELECT TEAMNO

FROM TEAMS

WHERE TRUE OR FALSE

Example 5.52:

SELECT PAYMENTNO, PAYMENTNO > 4

FROM PENALTIES

;

SELECT PAYMENTNO, CASE PAYMENTNO > 4

WHEN 1 THEN 'Greater than 4'

ELSE 'Less than 5'

END AS GREATER\_LESS

FROM PENALTIES

Example 5.53:

SELECT PLAYERNO, JOINED, PLAYERNO < 15, JOINED > 1979

FROM PLAYERS

WHERE (PLAYERNO < 15) = (JOINED > 1979)

Example 5.54:

INSERT INTO COMMITTEE\_MEMBERS

VALUES (7 + 15, CURRENT\_DATE,

CURRENT\_DATE + INTERVAL 17 DAY, 'Member')

Example 5.55:

SELECT PLAYERNO

FROM PLAYERS

WHERE (TOWN, STREET) = ('Stratford', 'Haseltine Lane')

Example 5.56:

SELECT PLAYERNO

FROM PLAYERS

WHERE (TOWN, STREET) = (SELECT 'Stratford', 'Haseltine Lane')

Example 5.57:

INSERT INTO PENALTIES VALUES

(1, 6, '1980-12-08', 100),

(2, 44, '1981-05-05', 75),

(3, 27, '1983-09-10', 100),

(4, 104, '1984-12-08', 50),

(5, 44, '1980-12-08', 25),

(6, 8, '1980-12-08', 25),

(7, 44, '1982-12-30', 30),

(8, 27, '1984-11-12', 75)

Answer 5.6:

SELECT PLAYERS.PLAYERNO, PLAYERS.NAME,

PLAYERS.INITIALS

FROM PLAYERS

WHERE PLAYERS.PLAYERNO > 6

ORDER BY PLAYERS.NAME

Answer 5.12:

SELECT PLAYERNO

FROM COMMITTEE\_MEMBERS

WHERE BEGIN\_DATE = CURRENT\_DATE

Answer 5.13:

SELECT TEAMNO,

CASE DIVISION

WHEN 'first' then 'first division'

WHEN 'second' THEN 'second division'

ELSE 'unknown' END AS DIVISION

FROM TEAMS

Answer 5.14:

SELECT PAYMENTNO, AMOUNT,

CASE

WHEN AMOUNT >= 0 AND AMOUNT <= 40

THEN 'low'

WHEN AMOUNT >= 41 AND AMOUNT <= 80

THEN 'moderate'

WHEN AMOUNT >= 81

THEN 'high'

ELSE 'incorrect' END AS CATEGORY

FROM PENALTIES

Answer 5.15:

SELECT PAYMENTNO, AMOUNT

FROM PENALTIES

WHERE CASE

WHEN AMOUNT >= 0 AND AMOUNT <= 40

THEN 'low'

WHEN AMOUNT > 40 AND AMOUNT <= 80

THEN 'moderate'

WHEN AMOUNT > 80

THEN 'high'

ELSE 'incorrect' END = 'low'

Answer 5.17:

SELECT PAYMENTNO

FROM PENALTIES

WHERE DAYNAME(PAYMENT\_DATE) = 'Monday'

Answer 5.18:

SELECT PAYMENTNO

FROM PENALTIES

WHERE YEAR(PAYMENT\_DATE) = 1984

Answer 5.25:

SELECT PLAYERNO, SUBSTR(INITIALS,1,1) || '. ' || NAME

FROM PLAYERS

Answer 5.26:

SELECT TEAMNO, RTRIM(DIVISION) || ' division'

FROM TEAMS

Answer 5.28:

SELECT PLAYERNO, BEGIN\_DATE,

BEGIN\_DATE + INTERVAL 2 MONTH + INTERVAL 3 DAY

FROM COMMITTEE\_MEMBERS

Answer 5.32:

SELECT PAYMENTNO, PAYMENT\_DATE,

PAYMENT\_DATE + INTERVAL 3 HOUR +

INTERVAL 50 SECOND + INTERVAL 99 MICROSECOND

FROM PENALTIES

Answer 5.33:

SELECT PLAYERNO,

CASE TOWN='Inglewood'

WHEN 1 THEN 'Yes' ELSE 'No' END

FROM PLAYERS

Answer 5.34:

SELECT \*

FROM PENALTIES

WHERE (AMOUNT = 25) = (PLAYERNO = 44)

Answer 5.35:

SELECT PAYMENTNO

FROM PENALTIES

WHERE (AMOUNT, PLAYERNO, PAYMENT\_DATE) =

(25, 44, '1980-12-08')

Answer 5.36:

SELECT PLAYERNO

FROM PLAYERS

WHERE (NAME, INITIALS) = (TOWN, STREET)

Answer 5.37:

SELECT \*

FROM PENALTIES

WHERE (AMOUNT = 25, PLAYERNO = 44) = (FALSE, TRUE)

Example 6.1:

SELECT PLAYERNO

FROM PENALTIES

WHERE AMOUNT > 25

GROUP BY PLAYERNO

HAVING COUNT(\*) > 1

ORDER BY PLAYERNO

Example 6.2:

SELECT PLAYERNO, LEAGUENO

FROM PLAYERS

WHERE TOWN = 'Stratford'

ORDER BY LEAGUENO

Example 6.3:

SELECT 89 \* 73

Example 6.4:

(SELECT \*

FROM TEAMS)

;

(((((SELECT \*

FROM TEAMS)))))

Example 6.5:

SELECT PLAYERNO

FROM TEAMS

UNION

SELECT PLAYERNO

FROM PENALTIES

;

SELECT PLAYERNO

FROM TEAMS

ORDER BY PLAYERNO

UNION

SELECT PLAYERNO

FROM PENALTIES

;

SELECT PLAYERNO

FROM TEAMS

UNION

SELECT PLAYERNO

FROM PENALTIES

ORDER BY PLAYERNO

;

(SELECT PLAYERNO

FROM TEAMS

ORDER BY PLAYERNO)

UNION

(SELECT PLAYERNO

FROM PENALTIES)

ORDER BY PLAYERNO

Example 6.6:

SELECT PLAYERNO

FROM (SELECT PLAYERNO, SEX

FROM PLAYERS

WHERE PLAYERNO < 10) AS PLAYERS10

WHERE SEX = 'M'

;

SELECT \*

FROM (SELECT \*

FROM (SELECT \*

FROM (SELECT \*

FROM PLAYERS) AS S1) AS S2) AS S3

Example 6.7:

SELECT PLAYERNO

FROM (SELECT PLAYERNO, SEX

FROM (SELECT PLAYERNO, SEX, JOINED

FROM (SELECT PLAYERNO, SEX, JOINED

FROM PLAYERS

WHERE PLAYERNO > 10) AS GREATER10

WHERE PLAYERNO < 100) AS LESS100

WHERE JOINED > 1980) AS JOINED1980

WHERE SEX = 'M'

Example 6.8:

SELECT PLAYERNO, JOINED -

(SELECT JOINED

FROM PLAYERS

WHERE PLAYERNO = 100)

FROM PLAYERS

WHERE PLAYERNO < 60

;

SELECT PLAYERNO, JOINED - 1979

FROM PLAYERS

WHERE PLAYERNO < 60

;

SELECT TEAMNO

FROM TEAMS

WHERE PLAYERNO =

(SELECT PLAYERNO

FROM PLAYERS)

Example 6.9:

SELECT PLAYERNO

FROM PLAYERS

WHERE YEAR(BIRTH\_DATE) = (SELECT YEAR(BIRTH\_DATE)

FROM PLAYERS

WHERE PLAYERNO = 27)

;

SELECT PLAYERNO

FROM PLAYERS

WHERE YEAR(BIRTH\_DATE) = 1964

Example 6.10:

SELECT (SELECT BIRTH\_DATE

FROM PLAYERS

WHERE PLAYERNO = 27),

(SELECT BIRTH\_DATE

FROM PLAYERS

WHERE PLAYERNO = 44),

(SELECT BIRTH\_DATE

FROM PLAYERS

WHERE PLAYERNO = 100)

Example 6.11:

SELECT PLAYERNO

FROM PLAYERS

WHERE (SEX, TOWN) = (SELECT SEX, TOWN

FROM PLAYERS

WHERE PLAYERNO = 100)

Answer 6.8:

SELECT PLAYERNO, BEGIN\_DATE

FROM COMMITTEE\_MEMBERS

UNION

SELECT PLAYERNO, END\_DATE

FROM COMMITTEE\_MEMBERS

ORDER BY PLAYERNO

Answer 6.9:

SELECT PLAYERNO, BEGIN\_DATE, 'Begin date'

FROM COMMITTEE\_MEMBERS

UNION

SELECT PLAYERNO, END\_DATE, 'End date'

FROM COMMITTEE\_MEMBERS

ORDER BY PLAYERNO

Answer 6.10:

SELECT PLAYERNO

FROM (SELECT PLAYERNO

FROM (SELECT PLAYERNO, END\_DATE

FROM (SELECT PLAYERNO, BEGIN\_DATE,

END\_DATE

FROM COMMITTEE\_MEMBERS

WHERE POSITION = 'Secretary')

AS SECRETARIES

WHERE BEGIN\_DATE >= '1990-01-01')

AS AFTER1989

WHERE END\_DATE <= '1994-12-31') AS BEFORE1995

Answer 6.11:

SELECT TEAMNO

FROM TEAMS

WHERE PLAYERNO =

(SELECT PLAYERNO

FROM PLAYERS

WHERE NAME = 'Parmenter'

AND INITIALS = 'R')

Answer 6.12:

SELECT TEAMNO

FROM TEAMS

WHERE PLAYERNO =

(SELECT PLAYERNO

FROM PLAYERS

WHERE NAME =

(SELECT NAME

FROM PLAYERS

WHERE PLAYERNO = 6)

AND PLAYERNO <> 6)

;

SELECT NAME

FROM PLAYERS

WHERE PLAYERNO =

(SELECT PLAYERNO

FROM TEAMS

WHERE TEAMNO =

(SELECT TEAMNO

FROM MATCHES

WHERE MATCHNO = 6))

Answer 6.13:

SELECT PAYMENTNO

FROM PENALTIES

WHERE AMOUNT >

(SELECT AMOUNT

FROM PENALTIES

WHERE PAYMENTNO = 4)

Answer 6.14:

SELECT PLAYERNO

FROM PLAYERS

WHERE DAYNAME(BIRTH\_DATE) =

(SELECT DAYNAME(BIRTH\_DATE)

FROM PLAYERS

WHERE PLAYERNO = 2)

Answer 6.15:

SELECT PLAYERNO

FROM COMMITTEE\_MEMBERS

WHERE (BEGIN\_DATE, END\_DATE) =

(SELECT BEGIN\_DATE, END\_DATE

FROM COMMITTEE\_MEMBERS

WHERE PLAYERNO = 8

AND POSITION = 'Treasurer')

AND PLAYERNO <> 8

Answer 6.16:

SELECT (SELECT DIVISION

FROM TEAMS

WHERE TEAMNO = 1),

(SELECT DIVISION

FROM TEAMS

WHERE TEAMNO = 2)

Answer 6.17:

SELECT (SELECT AMOUNT

FROM PENALTIES

WHERE PAYMENTNO = 1) +

(SELECT AMOUNT

FROM PENALTIES

WHERE PAYMENTNO = 2) +

(SELECT AMOUNT

FROM PENALTIES

WHERE PAYMENTNO = 3)

Example 7.1:

CREATE DATABASE EXTRA

;

USE EXTRA

;

CREATE TABLE CITIES

(CITYNO INTEGER NOT NULL PRIMARY KEY,

CITYNAME CHAR(20) NOT NULL)

;

INSERT INTO CITIES VALUES

(1, 'Stratford')

;

INSERT INTO CITIES VALUES

(2, 'Inglewood')

Example 7.2:

SELECT \*

FROM EXTRA.CITIES

Example 7.3:

SELECT \*

FROM TENNIS.TEAMS

Example 7.4:

SELECT TEAMNO

FROM TEAMS

;

SELECT TEAMS.TEAMNO

FROM TEAMS

;

SELECT TENNIS.TEAMS.TEAMNO

FROM TENNIS.TEAMS

Example 7.5:

SELECT TEAMNO, NAME

FROM TEAMS, PLAYERS

WHERE TEAMS.PLAYERNO = PLAYERS.PLAYERNO

Example 7.6:

SELECT PAYMENTNO, PENALTIES.PLAYERNO, AMOUNT,

NAME, INITIALS

FROM PENALTIES, PLAYERS

WHERE PENALTIES.PLAYERNO = PLAYERS.PLAYERNO

;

SELECT PLAYERS.PLAYERNO

FROM PLAYERS, TEAMS

WHERE PLAYERS.PLAYERNO = TEAMS.PLAYERNO

;

SELECT PLAYERS.PLAYERNO

FROM TEAMS, PLAYERS

WHERE PLAYERS.PLAYERNO = TEAMS.PLAYERNO

Example 7.7:

SELECT PAYMENTNO, PEN.PLAYERNO, AMOUNT,

NAME, INITIALS

FROM PENALTIES AS PEN, PLAYERS AS P

WHERE PEN.PLAYERNO = P.PLAYERNO

;

SELECT PAYMENTNO, PEN.PLAYERNO, AMOUNT,

NAME, INITIALS

FROM PENALTIES PEN, PLAYERS P

WHERE PEN.PLAYERNO = P.PLAYERNO

Example 7.8:

SELECT T.PLAYERNO

FROM TEAMS AS T, PENALTIES AS PEN

WHERE T.PLAYERNO = PEN.PLAYERNO

Example 7.9:

SELECT DISTINCT T.PLAYERNO

FROM TEAMS AS T, PENALTIES AS PEN

WHERE T.PLAYERNO = PEN.PLAYERNO

Example 7.10:

SELECT DISTINCT P.NAME, P.INITIALS

FROM PLAYERS AS P, MATCHES AS M

WHERE P.PLAYERNO = M.PLAYERNO

Example 7.11:

SELECT M.MATCHNO, M.PLAYERNO, M.TEAMNO, P.NAME, T.DIVISION

FROM MATCHES AS M, PLAYERS AS P, TEAMS AS T

WHERE M.PLAYERNO = P.PLAYERNO

AND M.TEAMNO = T.TEAMNO

Example 7.12:

SELECT PEN.PAYMENTNO, PEN.PLAYERNO, PEN.PAYMENT\_DATE

FROM PENALTIES AS PEN, PLAYERS AS P

WHERE PEN.PLAYERNO = P.PLAYERNO

AND YEAR(PEN.PAYMENT\_DATE) = P.JOINED

Example 7.13:

SELECT P.PLAYERNO

FROM PLAYERS AS P, PLAYERS AS PAR

WHERE PAR.NAME = 'Parmenter'

AND PAR.INITIALS = 'R'

AND P.BIRTH\_DATE < PAR.BIRTH\_DATE

;

SELECT P.PLAYERNO

FROM PLAYERS AS P, PLAYERS

WHERE PLAYERS.NAME = 'Parmenter'

AND PLAYERS.INITIALS = 'R'

AND P.BIRTH\_DATE < PLAYERS.BIRTH\_DATE

Example 7.14:

SELECT P.PLAYERNO

FROM PLAYERS AS P, EXTRA.CITIES AS TOWN

WHERE P.TOWN = TOWN.CITYNAME

Example 7.15:

SELECT PLAYERS.PLAYERNO, NAME, AMOUNT

FROM PLAYERS, PENALTIES

WHERE PLAYERS.PLAYERNO = PENALTIES.PLAYERNO

AND BIRTH\_DATE > '1920-06-30'

;

SELECT PLAYERS.PLAYERNO, NAME, AMOUNT

FROM PLAYERS INNER JOIN PENALTIES

ON (PLAYERS.PLAYERNO = PENALTIES.PLAYERNO)

WHERE BIRTH\_DATE > '1920-06-30'

Example 7.16:

SELECT TEAMNO, NAME

FROM TEAMS, PLAYERS

WHERE TEAMS.PLAYERNO = PLAYERS.PLAYERNO

;

SELECT TEAMNO, NAME

FROM TEAMS INNER JOIN PLAYERS

ON TEAMS.PLAYERNO = PLAYERS.PLAYERNO

;

SELECT TEAMNO, NAME

FROM TEAMS JOIN PLAYERS

ON TEAMS.PLAYERNO = PLAYERS.PLAYERNO

Example 7.17:

SELECT PLAYERS.PLAYERNO, NAME, AMOUNT

FROM PLAYERS, PENALTIES

WHERE PLAYERS.PLAYERNO = PENALTIES.PLAYERNO

ORDER BY PLAYERS.PLAYERNO

;

SELECT PLAYERS.PLAYERNO, NAME, AMOUNT

FROM PLAYERS LEFT OUTER JOIN PENALTIES

ON PLAYERS.PLAYERNO = PENALTIES.PLAYERNO

ORDER BY PLAYERS.PLAYERNO

Example 7.18:

SELECT PAYMENTNO, NAME

FROM PENALTIES LEFT OUTER JOIN PLAYERS

ON PENALTIES.PLAYERNO = PLAYERS.PLAYERNO

ORDER BY PAYMENTNO

Example 7.19:

SELECT P.PLAYERNO, NAME, TEAMNO, DIVISION

FROM PLAYERS AS P LEFT OUTER JOIN TEAMS AS T

ON P.PLAYERNO = T.PLAYERNO

ORDER BY P.PLAYERNO

Example 7.20:

SELECT PLAYERS.PLAYERNO, NAME, AMOUNT, TEAMNO

FROM PLAYERS LEFT OUTER JOIN PENALTIES

ON PLAYERS.PLAYERNO = PENALTIES.PLAYERNO

LEFT OUTER JOIN MATCHES

ON PLAYERS.PLAYERNO = MATCHES.PLAYERNO

WHERE TOWN = 'Inglewood'

Example 7.21:

SELECT PLAYERS.PLAYERNO, NAME, TEAMNO

FROM TEAMS RIGHT OUTER JOIN PLAYERS

ON TEAMS.PLAYERNO = PLAYERS.PLAYERNO

Example 7.22:

SELECT PLAYERS.PLAYERNO, NAME, AMOUNT

FROM PLAYERS INNER JOIN PENALTIES

ON PLAYERS.PLAYERNO = PENALTIES.PLAYERNO

WHERE BIRTH\_DATE > '1920-06-30'

;

SELECT PLAYERS.PLAYERNO, NAME, AMOUNT

FROM PLAYERS NATURAL JOIN PENALTIES

WHERE BIRTH\_DATE > '1920-06-30'

Example 7.23:

SELECT TEAMS.PLAYERNO, TEAMS.TEAMNO, PENALTIES.PAYMENTNO

FROM TEAMS LEFT OUTER JOIN PENALTIES

ON TEAMS.PLAYERNO = PENALTIES.PLAYERNO

WHERE DIVISION = 'second'

;

SELECT TEAMS.PLAYERNO, TEAMS.TEAMNO, PENALTIES.PAYMENTNO

FROM TEAMS LEFT OUTER JOIN PENALTIES

ON TEAMS.PLAYERNO = PENALTIES.PLAYERNO

AND DIVISION = 'second'

Example 7.24:

SELECT TEAMS.PLAYERNO, TEAMS.TEAMNO, PENALTIES.PAYMENTNO

FROM TEAMS FULL OUTER JOIN PENALTIES

ON TEAMS.PLAYERNO = PENALTIES.PLAYERNO

AND TEAMS.PLAYERNO > 1000

Example 7.25:

SELECT \*

FROM PENALTIES LEFT OUTER JOIN TEAMS

USING (PLAYERNO)

Example 7.26:

SELECT PLAYERNO

FROM (SELECT \*

FROM PLAYERS

WHERE TOWN = 'Stratford') AS STRATFORDERS

Example 7.27:

SELECT SMALL\_TEAMS.PLAYERNO

FROM (SELECT PLAYERNO, DIVISION

FROM TEAMS) AS SMALL\_TEAMS

WHERE SMALL\_TEAMS.DIVISION = 'first'

Example 7.28:

SELECT MATCHNO, DIFFERENCE

FROM (SELECT MATCHNO,

ABS(WON – LOST) AS DIFFERENCE

FROM MATCHES) AS M

WHERE DIFFERENCE > 2

Example 7.29:

SELECT \*

FROM (SELECT 'Stratford' AS TOWN, 4 AS NUMBER

UNION

SELECT 'Plymouth', 6

UNION

SELECT 'Inglewood', 1

UNION

SELECT 'Douglas', 2) AS TOWNS

ORDER BY TOWN

Example 7.30:

SELECT PLAYERNO, NAME, PLAYERS.TOWN, NUMBER \* 1000

FROM PLAYERS,

(SELECT 'Stratford' AS TOWN, 4 AS NUMBER

UNION

SELECT 'Plymouth', 6

UNION

SELECT 'Inglewood', 1

UNION

SELECT 'Douglas', 2) AS TOWNS

WHERE PLAYERS.TOWN = TOWNS.TOWN

ORDER BY PLAYERNO

;

SELECT PLAYERNO, NAME, PLAYERS.TOWN, NUMBER

FROM PLAYERS LEFT OUTER JOIN

(SELECT 'Stratford' AS TOWN, 4 AS NUMBER

UNION

SELECT 'Plymouth', 6

UNION

SELECT 'Inglewood', 1

UNION

SELECT 'Douglas', 2) AS TOWNS

ON PLAYERS.TOWN = TOWNS.TOWN

ORDER BY PLAYERNO

Example 7.31:

SELECT PLAYERNO

FROM PLAYERS LEFT OUTER JOIN

(SELECT 'Stratford' AS TOWN, 4 AS NUMBER

UNION

SELECT 'Plymouth', 6

UNION

SELECT 'Inglewood', 1

UNION

SELECT 'Douglas', 2) AS TOWNS

ON PLAYERS.TOWN = TOWNS.TOWN

WHERE TOWNS.NUMBER > 2

Example 7.32:

SELECT \*

FROM (SELECT 'John' AS FIRST\_NAME

UNION

SELECT 'Mark'

UNION

SELECT 'Arnold') AS FIRST\_NAMES,

(SELECT 'Berg' AS LAST\_NAME

UNION

SELECT 'Johnson'

UNION

SELECT 'Williams') AS LAST\_NAMES

Example 7.33:

SELECT NUMBER, POWER(NUMBER,3)

FROM (SELECT 10 AS NUMBER UNION SELECT 11 UNION SELECT 12

UNION

SELECT 13 UNION SELECT 14 UNION SELECT 15

UNION

SELECT 16 UNION SELECT 17 UNION SELECT 18

UNION

SELECT 19) AS NUMBERS

WHERE POWER(NUMBER,3) <= 4000

Example 7.34:

SELECT NUMBER

FROM (SELECT CAST(CONCAT(DIGIT1.DIGIT,

CONCAT(DIGIT2.DIGIT,

DIGIT3.DIGIT)) AS UNSIGNED INTEGER)

AS NUMBER

FROM (SELECT '0' AS DIGIT UNION SELECT '1' UNION

SELECT '2' UNION SELECT '3' UNION

SELECT '4' UNION SELECT '5' UNION

SELECT '6' UNION SELECT '7' UNION

SELECT '8' UNION SELECT '9') AS DIGIT1,

(SELECT '0' AS DIGIT UNION SELECT '1' UNION

SELECT '2' UNION SELECT '3' UNION

SELECT '4' UNION SELECT '5' UNION

SELECT '6' UNION SELECT '7' UNION

SELECT '8' UNION SELECT '9') AS DIGIT2,

(SELECT '0' AS DIGIT UNION SELECT '1' UNION

SELECT '2' UNION SELECT '3' UNION

SELECT '4' UNION SELECT '5' UNION

SELECT '6' UNION SELECT '7' UNION

SELECT '8' UNION SELECT '9') AS DIGIT3)

AS NUMBERS

ORDER BY NUMBER

Example 7.35:

SELECT NUMBER AS SQUARE, ROUND(SQRT(NUMBER)) AS BASIS

FROM (SELECT CAST(CONCAT(DIGIT1.DIGIT,

CONCAT(DIGIT2.DIGIT,

DIGIT3.DIGIT)) AS UNSIGNED INTEGER)

AS NUMBER

FROM (SELECT '0' AS DIGIT UNION SELECT '1' UNION

SELECT '2' UNION SELECT '3' UNION

SELECT '4' UNION SELECT '5' UNION

SELECT '6' UNION SELECT '7' UNION

SELECT '8' UNION SELECT '9') AS DIGIT1,

(SELECT '0' AS DIGIT UNION SELECT '1' UNION

SELECT '2' UNION SELECT '3' UNION

SELECT '4' UNION SELECT '5' UNION

SELECT '6' UNION SELECT '7' UNION

SELECT '8' UNION SELECT '9') AS DIGIT2,

(SELECT '0' AS DIGIT UNION SELECT '1' UNION

SELECT '2' UNION SELECT '3' UNION

SELECT '4' UNION SELECT '5' UNION

SELECT '6' UNION SELECT '7' UNION

SELECT '8' UNION SELECT '9') AS DIGIT3)

AS NUMBERS

WHERE SQRT(NUMBER) = ROUND(SQRT(NUMBER))

ORDER BY NUMBER

Answer 7.3:

SELECT PAYMENTNO, AMOUNT, PLAYERS.PLAYERNO, NAME

FROM PENALTIES, PLAYERS

WHERE PENALTIES.PLAYERNO = PLAYERS.PLAYERNO

Answer 7.4:

SELECT PAYMENTNO, NAME

FROM PENALTIES, PLAYERS, TEAMS

WHERE PENALTIES.PLAYERNO = TEAMS.PLAYERNO

AND TEAMS.PLAYERNO = PLAYERS.PLAYERNO

Answer 7.5:

SELECT T.TEAMNO, P.NAME

FROM TEAMS AS T, PLAYERS AS P

WHERE T.PLAYERNO = P.PLAYERNO

Answer 7.6:

SELECT M.MATCHNO, P.NAME, T.DIVISION

FROM MATCHES AS M, PLAYERS AS P, TEAMS AS T

WHERE M.PLAYERNO = P.PLAYERNO

AND M.TEAMNO = T.TEAMNO

Answer 7.7:

SELECT P.PLAYERNO, P.NAME

FROM PLAYERS AS P, COMMITTEE\_MEMBERS AS C

WHERE P.PLAYERNO = C.PLAYERNO

AND B.POSITION = 'Chairman'

Answer 7.8:

SELECT DISTINCT CM.PLAYERNO

FROM COMMITTEE\_MEMBERS AS CM, PENALTIES AS PEN

WHERE CM.PLAYERNO = PEN.PLAYERNO

AND CM.BEGIN\_DATE = PEN.PAYMENT\_DATE

Answer 7.9:

SELECT P.PLAYERNO, P.NAME

FROM PLAYERS AS P, PLAYERS AS P27

WHERE P.TOWN = P27.TOWN

AND P27.PLAYERNO = 27

AND P.PLAYERNO <> 27

Answer 7.10:

SELECT DISTINCT P.PLAYERNO AS PLAYER\_PLAYERNO,

P.NAME AS PLAYER\_NAME,

CAP.PLAYERNO AS CAPTAIN\_PLAYERNO,

CAP.NAME AS CAPTAIN\_NAME

FROM PLAYERS AS P, PLAYERS AS CAP,

MATCHES AS M, TEAMS AS T

WHERE M.PLAYERNO = P.PLAYERNO

AND T.TEAMNO = M.TEAMNO

AND M.PLAYERNO <> T.PLAYERNO

AND CAP.PLAYERNO = T.PLAYERNO

Answer 7.11:

SELECT PEN1.PAYMENTNO, PEN1.PLAYERNO

FROM PENALTIES AS PEN1, PENALTIES AS PEN2

WHERE PEN1.AMOUNT = PEN2.AMOUNT

AND PEN2.PLAYERNO = 44

AND PEN1.PLAYERNO <> 44

Answer 7.12:

SELECT T.TEAMNO, P.NAME

FROM TEAMS AS T INNER JOIN PLAYERS AS P

ON T.PLAYERNO = P.PLAYERNO

Answer 7.13:

SELECT P.PLAYERNO, P.NAME

FROM PLAYERS AS P INNER JOIN PLAYERS AS P27

ON P.TOWN = P27.TOWN

AND P27.PLAYERNO = 27

AND P.PLAYERNO <> 27

Answer 7.14:

SELECT M.MATCHNO, P.NAME, T.DIVISION

FROM (MATCHES AS M INNER JOIN PLAYERS AS P

ON M.PLAYERNO = P.PLAYERNO)

INNER JOIN TEAMS AS T

ON M.TEAMNO = T.TEAMNO

Answer 7.15:

SELECT PLAYERS.PLAYERNO, PENALTIES.AMOUNT

FROM PLAYERS LEFT OUTER JOIN PENALTIES

ON PLAYERS.PLAYERNO = PENALTIES.PLAYERNO

Answer 7.16:

SELECT P.PLAYERNO, M.TEAMNO

FROM PLAYERS AS P LEFT OUTER JOIN MATCHES AS M

ON P.PLAYERNO = M.PLAYERNO

Answer 7.17:

SELECT P.PLAYERNO, PEN.AMOUNT, M.TEAMNO

FROM (PLAYERS AS P LEFT OUTER JOIN MATCHES AS M

ON P.PLAYERNO = M.PLAYERNO)

LEFT OUTER JOIN PENALTIES AS PEN

ON P.PLAYERNO = PEN.PLAYERNO

Answer 7.21:

SELECT PLAYERNO, DIFFERENCE

FROM (SELECT PLAYERNO,

JOINED - YEAR(BIRTH\_DATE) AS DIFFERENCE

FROM PLAYERS) AS DIFFERENCES

WHERE DIFFERENCE > 20

Answer 7.22:

SELECT LETTER1 || LETTER2 || LETTER3

FROM (SELECT 'a' AS LETTER1 UNION SELECT 'b'

UNION SELECT 'c' UNION SELECT 'd') AS LETTERS1,

(SELECT 'a' AS LETTER2 UNION SELECT 'b'

UNION SELECT 'c' UNION SELECT 'd') AS LETTERS2,

(SELECT 'a' AS LETTER3 UNION SELECT 'b'

UNION SELECT 'c' UNION SELECT 'd') AS LETTERS3

Answer 7.23:

SELECT ROUND(RAND() \* 1000)

FROM (SELECT 0 AS NUMBER UNION SELECT 1 UNION SELECT 2

UNION

SELECT 3 UNION SELECT 4 UNION SELECT 5

UNION

SELECT 6 UNION SELECT 7 UNION SELECT 8

UNION

SELECT 9) AS NUMBERS

Example 8.1:

SELECT PLAYERNO

FROM PLAYERS

WHERE TOWN = 'Stratford'

Example 8.2:

SELECT PLAYERNO, BIRTH\_DATE, JOINED

FROM PLAYERS

WHERE YEAR(BIRTH\_DATE) + 17 = JOINED

Example 8.3:

SELECT PLAYERNO

FROM PLAYERS

WHERE LEAGUENO = '7060'

Example 8.4:

SELECT PLAYERNO, LEAGUENO

FROM PLAYERS

WHERE LEAGUENO = LEAGUENO

Example 8.5:

SELECT PLAYERNO, LEAGUENO

FROM PLAYERS

WHERE LEAGUENO <=> NULL

Example 8.6:

SELECT MATCHNO

FROM MATCHES

WHERE (WON, LOST) = (2, 3)

Example 8.7:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE PLAYERNO =

(SELECT PLAYERNO

FROM TEAMS

WHERE TEAMNO = 1)

;

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE PLAYERNO = 6

;

SELECT \*

FROM PLAYERS

WHERE BIRTH\_DATE <

(SELECT BIRTH\_DATE

FROM PLAYERS)

Example 8.8:

SELECT PLAYERNO, NAME, INITIALS

FROM PLAYERS

WHERE BIRTH\_DATE <

(SELECT BIRTH\_DATE

FROM PLAYERS

WHERE LEAGUENO = '8467')

;

SELECT PLAYERNO, NAME, INITIALS

FROM PLAYERS

WHERE BIRTH\_DATE <

(SELECT BIRTH\_DATE

FROM PLAYERS

WHERE LEAGUENO = '9999')

Example 8.9:

SELECT MATCHNO

FROM MATCHES

WHERE TEAMNO =

(SELECT TEAMNO

FROM TEAMS

WHERE PLAYERNO = 27)

Example 8.10:

SELECT PLAYERNO

FROM PLAYERS

WHERE LEAGUENO <=>

(SELECT LEAGUENO

FROM PLAYERS

WHERE PLAYERNO = 7)

Example 8.11:

SELECT PLAYERNO, TOWN, SEX

FROM PLAYERS

WHERE (TOWN, SEX) =

((SELECT TOWN

FROM PLAYERS

WHERE PLAYERNO = 7),

(SELECT SEX

FROM PLAYERS

WHERE PLAYERNO = 2))

Example 8.12:

SELECT DISTINCT PLAYERNO

FROM COMMITTEE\_MEMBERS

WHERE (BEGIN\_DATE, END\_DATE) =

(SELECT BEGIN\_DATE, END\_DATE

FROM COMMITTEE\_MEMBERS

WHERE PLAYERNO = 6

AND POSITION = 'Secretary'

AND BEGIN\_DATE = '1990-01-01')

Example 8.13:

SELECT PLAYERNO, NAME, INITIALS

FROM PLAYERS

WHERE (NAME, INITIALS) <

(SELECT NAME, INITIALS

FROM PLAYERS

WHERE PLAYERNO = 6)

ORDER BY NAME, INITIALS

Example 8.14:

SELECT MATCHNO

FROM MATCHES\_SPECIAL

WHERE (START\_DATE, START\_TIME) >

(SELECT START\_DATE, START\_TIME

FROM MATCHES\_SPECIAL

WHERE MATCHNO = 1)

Example 8.15:

SELECT MATCHNO

FROM MATCHES

WHERE 'Inglewood' =

(SELECT TOWN

FROM PLAYERS

WHERE PLAYERS.PLAYERNO = MATCHES.PLAYERNO)

;

SELECT TOWN

FROM PLAYERS

WHERE PLAYERS.PLAYERNO = 6

;

SELECT TOWN

FROM PLAYERS

WHERE PLAYERS.PLAYERNO = 44

Example 8.16:

SELECT MATCHNO, PLAYERNO, TEAMNO

FROM MATCHES

WHERE PLAYERNO =

(SELECT PLAYERNO

FROM TEAMS

WHERE TEAMS.PLAYERNO = MATCHES.PLAYERNO)

Example 8.17:

SELECT MATCHNO

FROM MATCHES

WHERE SUBSTR((SELECT DIVISION

FROM TEAMS

WHERE TEAMS.TEAMNO =

MATCHES.TEAMNO),3,1)

=

SUBSTR((SELECT NAME

FROM PLAYERS

WHERE PLAYERS.PLAYERNO =

MATCHES.PLAYERNO),3,1)

Example 8.18:

SELECT PLAYERNO, LEAGUENO

FROM PLAYERS

WHERE LEAGUENO

Example 8.19:

SELECT TEAMNO

FROM TEAMS

WHERE TEAMNO - 1

;

SELECT \* FROM PLAYERS WHERE 18

;

SELECT \* FROM PLAYERS WHERE NULL

;

SELECT \* FROM PLAYERS WHERE PLAYERNO & 3

;

SELECT \* FROM PLAYERS WHERE YEAR(BIRTH\_DATE)

Example 8.20:

SELECT PLAYERNO, NAME, SEX, BIRTH\_DATE

FROM PLAYERS

WHERE SEX = 'M'

AND BIRTH\_DATE > '1970-12-31'

Example 8.21:

SELECT PLAYERNO, NAME, TOWN

FROM PLAYERS

WHERE TOWN = 'Plymouth'

OR TOWN = 'Eltham'

Example 8.22:

SELECT PLAYERNO, NAME, TOWN

FROM PLAYERS

WHERE TOWN <> 'Stratford'

;

SELECT PLAYERNO, NAME, TOWN

FROM PLAYERS

WHERE NOT (TOWN = 'Stratford')

Example 8.23:

SELECT PLAYERNO, LEAGUENO, PHONENO

FROM PLAYERS

WHERE LEAGUENO AND PHONENO

Example 8.24:

SELECT PLAYERNO, TOWN, BIRTH\_DATE

FROM PLAYERS

WHERE (TOWN = 'Stratford' OR YEAR(BIRTH\_DATE) = 1963)

AND NOT (TOWN = 'Stratford' AND YEAR(BIRTH\_DATE) = 1963)

;

SELECT PLAYERNO, TOWN, BIRTH\_DATE

FROM PLAYERS

WHERE (TOWN = 'Stratford') XOR (YEAR(BIRTH\_DATE) = 1963)

Example 8.25:

SELECT PLAYERNO, NAME, TOWN

FROM PLAYERS

WHERE TOWN = 'Inglewood'

OR TOWN = 'Plymouth'

OR TOWN = 'Midhurst'

OR TOWN = 'Douglas'

;

SELECT PLAYERNO, NAME, TOWN

FROM PLAYERS

WHERE TOWN IN ('Inglewood', 'Plymouth', 'Midhurst',

'Douglas')

Example 8.26:

SELECT PLAYERNO, YEAR(BIRTH\_DATE)

FROM PLAYERS

WHERE YEAR(BIRTH\_DATE) IN (1962, 1963, 1970)

Example 8.27:

SELECT MATCHNO, WON, LOST

FROM MATCHES

WHERE 2 IN (WON, LOST)

Example 8.28:

SELECT PLAYERNO

FROM PLAYERS

WHERE PLAYERNO IN

(100,

(SELECT PLAYERNO

FROM PENALTIES

WHERE PAYMENTNO = 1),

(SELECT PLAYERNO

FROM TEAMS

WHERE TEAMNO = 2))

Example 8.29:

SELECT MATCHNO, WON, LOST

FROM MATCHES

WHERE WON IN

(TRUNCATE(MATCHNO / 2,0), LOST,

(SELECT LOST

FROM MATCHES

WHERE MATCHNO = 1))

Example 8.30:

SELECT MATCHNO

FROM MATCHES

WHERE (SELECT SUBSTR(NAME,1,1)

FROM PLAYERS

WHERE PLAYERS.PLAYERNO = MATCHES.PLAYERNO)

IN ('B','C','E')

Example 8.31:

SELECT MATCHNO, WON, LOST

FROM MATCHES

WHERE (WON, LOST) IN ((3,1),(3,2))

Example 8.32:

SELECT PLAYERNO, NAME, INITIALS

FROM PLAYERS

WHERE (NAME, INITIALS) IN

((SELECT NAME, INITIALS

FROM PLAYERS

WHERE PLAYERNO = 6),

(SELECT NAME, INITIALS

FROM PLAYERS

WHERE PLAYERNO = 27))

Example 8.33:

SELECT PLAYERNO

FROM MATCHES

;

SELECT PLAYERNO, NAME, INITIALS

FROM PLAYERS

WHERE PLAYERNO IN (6, 6, 6, 44, 83, 2, 57, 8, 27,

104, 112, 112, 8)

;

SELECT PLAYERNO, NAME, INITIALS

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM MATCHES)

;

SELECT PLAYERNO, NAME, INITIALS

FROM PLAYERS

WHERE PLAYERNO IN (6, 6, 6, 44, 83, 2, 57, 8, 27,

104, 112, 112, 8)

Example 8.34:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM MATCHES

WHERE TEAMNO = 1)

Example 8.35:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM MATCHES

WHERE TEAMNO NOT IN

(SELECT TEAMNO

FROM TEAMS

WHERE PLAYERNO = 6))

;

SELECT \*

FROM PLAYERS

WHERE BIRTH\_DATE NOT IN

(SELECT BIRTH\_DATE

FROM PLAYERS

WHERE Town = 'London')

Example 8.36:

SELECT \*

FROM COMMITTEE\_MEMBERS

WHERE (BEGIN\_DATE, END\_DATE) IN

(SELECT BEGIN\_DATE, END\_DATE

FROM COMMITTEE\_MEMBERS

WHERE POSITION = 'Secretary')

Example 8.37:

CREATE TABLE PLAYERS\_NI

(NAME CHAR(10) NOT NULL,

INITIALS CHAR(3) NOT NULL,

TOWN VARCHAR(30) NOT NULL,

PRIMARY KEY (NAME, INITIALS))

;

INSERT INTO PLAYERS\_NI VALUES ('Parmenter', 'R', 'Stratford')

;

INSERT INTO PLAYERS\_NI VALUES ('Parmenter', 'P', 'Stratford')

;

INSERT INTO PLAYERS\_NI VALUES ('Miller', 'P', 'Douglas')

;

CREATE TABLE PENALTIES\_NI

(PAYMENTNO INTEGER NOT NULL,

NAME CHAR(10) NOT NULL,

INITIALS CHAR(3) NOT NULL,

AMOUNT DECIMAL(7,2) NOT NULL,

PRIMARY KEY (PAYMENTNO),

FOREIGN KEY (NAME, INITIALS)

REFERENCES PLAYERS\_NI (NAME, INITIALS))

;

INSERT INTO PENALTIES\_NI VALUES (1, 'Parmenter', 'R', 100.00)

;

INSERT INTO PENALTIES\_NI VALUES (2, 'Miller', 'P', 200.00)

Example 8.38:

SELECT NAME, INITIALS, TOWN

FROM PLAYERS\_NI

WHERE NAME IN

(SELECT NAME

FROM PENALTIES\_NI)

AND INITIALS IN

(SELECT INITIALS

FROM PENALTIES\_NI)

;

SELECT NAME, INITIALS, TOWN

FROM PLAYERS\_NI

WHERE (NAME, INITIALS) IN

(SELECT NAME, INITIALS

FROM PENALTIES\_NI)

;

SELECT NAME, INITIALS, TOWN

FROM PLAYERS\_NI

WHERE NAME IN

(SELECT NAME

FROM PENALTIES\_NI

WHERE PLAYERS\_NI.INITIALS =

PENALTIES\_NI.INITIALS)

Example 8.39:

SELECT NAME, INITIALS, TOWN

FROM PLAYERS\_NI

WHERE (NAME, INITIALS) NOT IN

(SELECT NAME, INITIALS

FROM PENALTIES\_NI)

Example 8.40:

SELECT PLAYERNO, BIRTH\_DATE

FROM PLAYERS

WHERE BIRTH\_DATE >= '1962-01-01'

AND BIRTH\_DATE <= '1964-12-31'

;

SELECT PLAYERNO, BIRTH\_DATE

FROM PLAYERS

WHERE BIRTH\_DATE BETWEEN '1962-01-01' AND '1964-12-31'

Example 8.41:

SELECT MATCHNO, WON + LOST

FROM MATCHES

WHERE WON + LOST BETWEEN 2 AND 4

Example 8.42:

SELECT PLAYERNO, BIRTH\_DATE, NAME, INITIALS

FROM PLAYERS

WHERE BIRTH\_DATE BETWEEN

(SELECT BIRTH\_DATE

FROM PLAYERS

WHERE NAME = 'Newcastle'

AND INITIALS = 'B')

AND

(SELECT BIRTH\_DATE

FROM PLAYERS

WHERE NAME = 'Miller'

AND INITIALS = 'P')

Example 8.43:

SELECT NAME, PLAYERNO

FROM PLAYERS

WHERE NAME LIKE 'B%'

Example 8.44:

SELECT NAME, PLAYERNO

FROM PLAYERS

WHERE NAME LIKE '%r'

Example 8.45:

SELECT NAME, PLAYERNO

FROM PLAYERS

WHERE NAME LIKE '%e\_'

Example 8.46:

SELECT NAME, TOWN, PLAYERNO

FROM PLAYERS

WHERE NAME LIKE CONCAT('%', SUBSTR(TOWN,3,1))

Example 8.47:

SELECT NAME, PLAYERNO

FROM PLAYERS

WHERE NAME LIKE '%#\_%' ESCAPE '#'

Example 8.48:

SELECT NAME, PLAYERNO

FROM PLAYERS

WHERE NAME REGEXP 'e'

Example 8.49:

SELECT NAME, PLAYERNO

FROM PLAYERS

WHERE NAME REGEXP '^ba'

Example 8.50:

SELECT NAME, STREET, PLAYERNO

FROM PLAYERS

WHERE NAME REGEXP CONCAT(SUBSTR(STREET,1,1), '$')

Example 8.51:

SELECT NAME, PLAYERNO

FROM PLAYERS

WHERE NAME REGEXP '[abc]'

Example 8.52:

SELECT NAME, PLAYERNO

FROM PLAYERS

WHERE NAME REGEXP 'm.n'

Example 8.53:

SELECT NAME, PLAYERNO

FROM PLAYERS

WHERE NAME REGEXP '[men][men]'

Example 8.54:

SELECT PLAYERNO, POSTCODE

FROM PLAYERS

WHERE POSTCODE REGEXP '^[0-9][0-9]3'

Example 8.55:

SELECT STREET, PLAYERNO

FROM PLAYERS

WHERE STREET REGEXP '^St.\*Road$'

Example 8.56:

SELECT PLAYERNO, POSTCODE

FROM PLAYERS

WHERE POSTCODE REGEXP '[0-9][0-9]\*[a-z][a-z]\*'

Example 8.57:

SELECT NAME, PLAYERNO

FROM PLAYERS

WHERE NAME REGEXP '^[^A-M]'

Example 8.58:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE NAME REGEXP '^[a-z]{7}'

Example 8.59:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE NAME REGEXP '^[a-z]{6,7}$'

Example 8.60:

SELECT PLAYERNO, POSTCODE

FROM PLAYERS

WHERE POSTCODE REGEXP '4{4}'

Example 8.61:

SELECT PLAYERNO, STREET

FROM PLAYERS

WHERE STREET REGEXP 'Street|Square'

Example 8.62:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE NAME REGEXP '[[.space.]]'

Example 8.63:

SELECT PLAYERNO, STREET

FROM PLAYERS

WHERE STREET REGEXP '[[:<:]]Street[[:>:]]'

Example 8.64:

CREATE TABLE BOOKS

(BOOKNO INTEGER NOT NULL PRIMARY KEY,

AUTHORS TEXT NOT NULL,

TITLE TEXT NOT NULL,

YEAR\_PUBLICATION YEAR NOT NULL,

SUMMARY TEXT NOT NULL)

ENGINE = MyISAM

Example 8.65:

SET @@SQL\_MODE = 'PIPES\_AS\_CONCAT'

;

INSERT INTO BOOKS VALUES (1,

'Ramez Elmasri and Shamkant B. Navathe',

'Fundamentals of Database Systems', 2007,

'This market-leading text serves as a valued resource for '||

'those who will interact with databases in future courses '||

'and careers. Renowned for its accessible, comprehensive '||

'coverage of models and real systems, it provides an '||

'up-to-date introduction to modern database technologies.')

;

INSERT INTO BOOKS VALUES (2,

'George Coulouris, Jean Dollimore and Tim Kindberg',

'Distributed Systems: Concepts and Design', 2005,

'This book provides broad and up-to-date coverage of the '||

'principles and practice in the fast moving area of '||

'distributed systems. It includes the key issues in the '||

'debate between components and web services as the way '||

'forward for industry. The depth of coverage will enable '||

'students to evaluate existing distributed systems and '||

'design new ones.')

;

INSERT INTO BOOKS VALUES (3,

'Rick van der Lans',

'Introduction to SQL: Mastering the Relational Database '||

'Language', 2007,

'This book provides a technical introduction to the '||

'features of SQL. Aimed at those new to SQL, but not new '||

'to programming, it gives the reader the essential skills '||

'required to start programming with this language.')

;

INSERT INTO BOOKS VALUES (4,

'Chris Date',

'An Introduction to Database Systems', 2004,

'Continuing in the eighth edition, this book provides a '||

'comprehensive introduction to the now very large field of '||

'database systems by providing a solid grounding in the '||

'foundations of database technology. This new edition has '||

'been rewritten and expanded to stay current with database '||

'system trends.')

;

INSERT INTO BOOKS VALUES (5,

'Thomas M. Connolly and Carolyn E. Begg',

'DataBase Systems: A Practical Approach to Design, '||

'Implementation and Management',

2005,

'A clear introduction to design implementation and management '||

'issues, as well as an extensive treatment of database '||

'languages and standards, make this book an indispensable '||

'complete reference for database students and professionals.')

Example 8.66:

CREATE FULLTEXT INDEX INDEX\_TITLE

ON BOOKS (TITLE)

;

CREATE FULLTEXT INDEX INDEX\_SUMMARY

ON BOOKS (SUMMARY)

Example 8.67:

SELECT BOOKNO, TITLE

FROM BOOKS

WHERE MATCH(TITLE) AGAINST ('design')

;

SELECT BOOKNO, TITLE

FROM BOOKS

WHERE MATCH(TITLE)

AGAINST ('design' IN NATURAL LANGUAGE MODE)

Example 8.68:

SELECT BOOKNO, TITLE

FROM BOOKS

WHERE MATCH(TITLE) AGAINST ('to')

Example 8.69:

SELECT BOOKNO, TITLE

FROM BOOKS

WHERE MATCH(TITLE) AGAINST ('database')

Example 8.70:

SELECT BOOKNO, TITLE

FROM BOOKS

WHERE MATCH(TITLE) AGAINST ('practical')

Example 8.71:

SELECT BOOKNO, MATCH(SUMMARY) AGAINST ('distributed')

FROM BOOKS

Example 8.72:

SELECT BOOKNO, MATCH(TITLE) AGAINST ('introduction')

FROM BOOKS

WHERE MATCH(TITLE) AGAINST ('introduction')

Example 8.73:

SELECT BOOKNO, TITLE

FROM BOOKS

WHERE MATCH(TITLE) AGAINST ('practical distributed')

Example 8.74:

CREATE FULLTEXT INDEX INDEX\_TITLE\_SUMMARY

ON BOOKS (TITLE, SUMMARY)

Example 8.75:

SELECT BOOKNO, TITLE

FROM BOOKS

WHERE MATCH(TITLE, SUMMARY) AGAINST ('careers')

Example 8.76:

SELECT BOOKNO, TITLE

FROM BOOKS

WHERE MATCH(TITLE) AGAINST ('database' IN BOOLEAN MODE)

Example 8.77:

SELECT BOOKNO, TITLE

FROM BOOKS

WHERE MATCH(TITLE, SUMMARY)

AGAINST ('introduction' IN BOOLEAN MODE)

Example 8.78:

SELECT BOOKNO, TITLE

FROM BOOKS

WHERE MATCH(TITLE)

AGAINST ('database design' IN BOOLEAN MODE)

Example 8.79:

SELECT BOOKNO, TITLE

FROM BOOKS

WHERE MATCH(TITLE)

AGAINST ('+database +design' IN BOOLEAN MODE)

Example 8.80:

SELECT BOOKNO, TITLE

FROM BOOKS

WHERE MATCH(TITLE)

AGAINST ('+database -design' IN BOOLEAN MODE)

Example 8.81:

SELECT BOOKNO, TITLE

FROM BOOKS

WHERE MATCH(TITLE)

AGAINST ('"design implementation"' IN BOOLEAN MODE)

Example 8.82:

SELECT BOOKNO, TITLE

FROM BOOKS

WHERE MATCH(TITLE) AGAINST ('data\*' IN BOOLEAN MODE)

Example 8.83:

SELECT BOOKNO, TITLE

FROM BOOKS

WHERE MATCH(TITLE) AGAINST ('practical'

IN NATURAL LANGUAGE MODE WITH QUERY EXPANSION)

;

SELECT BOOKNO, TITLE

FROM BOOKS

WHERE MATCH(TITLE) AGAINST ('practical')

;

SELECT BOOKNO, TITLE

FROM BOOKS

WHERE MATCH(TITLE) AGAINST (' DataBase Systems: A Practical

Approach to Design, Implementation and Management')

Example 8.84:

SELECT BOOKNO, TITLE

FROM BOOKS

WHERE MATCH(TITLE) AGAINST ('sql')

Example 8.85:

SELECT PLAYERNO, LEAGUENO

FROM PLAYERS

WHERE LEAGUENO IS NOT NULL

Example 8.86:

SELECT NAME, PLAYERNO, LEAGUENO

FROM PLAYERS

WHERE LEAGUENO <> '8467'

OR LEAGUENO IS NULL

Example 8.87:

SELECT NAME, INITIALS

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PENALTIES)

;

SELECT NAME, INITIALS

FROM PLAYERS

WHERE EXISTS

(SELECT \*

FROM PENALTIES

WHERE PLAYERNO = PLAYERS.PLAYERNO)

;

SELECT \*

FROM PENALTIES

WHERE PLAYERNO = 6

Example 8.88:

SELECT NAME, INITIALS

FROM PLAYERS

WHERE NOT EXISTS

(SELECT \*

FROM TEAMS

WHERE PLAYERNO = PLAYERS.PLAYERNO)

;

SELECT NAME, INITIALS

FROM PLAYERS

WHERE NOT EXISTS

(SELECT 'nothing'

FROM TEAMS

WHERE PLAYERNO = PLAYERS.PLAYERNO)

Example 8.89:

SELECT PLAYERNO, NAME, BIRTH\_DATE

FROM PLAYERS

WHERE BIRTH\_DATE <= ALL

(SELECT BIRTH\_DATE

FROM PLAYERS)

Example 8.90:

SELECT PLAYERNO, BIRTH\_DATE

FROM PLAYERS

WHERE BIRTH\_DATE < ALL

(SELECT BIRTH\_DATE

FROM PLAYERS AS P INNER JOIN MATCHES AS M

ON P.PLAYERNO = M.PLAYERNO

WHERE M.TEAMNO = 2)

Example 8.91:

SELECT DISTINCT TEAMNO, PLAYERNO

FROM MATCHES AS M1

WHERE WON <= ALL

(SELECT WON

FROM MATCHES AS M2

WHERE M1.TEAMNO = M2.TEAMNO)

Example 8.92:

SELECT LEAGUENO, PLAYERNO

FROM PLAYERS

WHERE LEAGUENO >= ALL

(SELECT LEAGUENO

FROM PLAYERS)

;

SELECT LEAGUENO, PLAYERNO

FROM PLAYERS

WHERE LEAGUENO >= ALL

(SELECT LEAGUENO

FROM PLAYERS

WHERE LEAGUENO IS NOT NULL)

Example 8.93:

SELECT PLAYERNO, TOWN, LEAGUENO

FROM PLAYERS AS P1

WHERE LEAGUENO <= ALL

(SELECT P2.LEAGUENO

FROM PLAYERS AS P2

WHERE P1.TOWN = P2.TOWN)

;

SELECT PLAYERNO, TOWN, LEAGUENO

FROM PLAYERS AS P1

WHERE LEAGUENO <= ALL

(SELECT P2.LEAGUENO

FROM PLAYERS AS P2

WHERE P1.TOWN = P2.TOWN

AND LEAGUENO IS NOT NULL)

Example 8.94:

SELECT PLAYERNO, NAME, BIRTH\_DATE

FROM PLAYERS

WHERE BIRTH\_DATE > ANY

(SELECT BIRTH\_DATE

FROM PLAYERS)

Example 8.95:

SELECT DISTINCT PLAYERNO

FROM PENALTIES

WHERE PLAYERNO <> 27

AND AMOUNT > ANY

(SELECT AMOUNT

FROM PENALTIES

WHERE PLAYERNO = 27)

Example 8.96:

SELECT PLAYERNO, BIRTH\_DATE, TOWN

FROM PLAYERS AS P1

WHERE BIRTH\_DATE > ANY

(SELECT BIRTH\_DATE

FROM PLAYERS AS P2

WHERE P1.TOWN = P2.TOWN)

Example 8.98:

SELECT TEAMNO, DIVISION

FROM TEAMS

WHERE EXISTS

(SELECT \*

FROM MATCHES

WHERE PLAYERNO = 44

AND TEAMNO = TEAMS.TEAMNO)

Example 8.99:

SELECT DISTINCT PLAYERNO

FROM PENALTIES AS PEN

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PENALTIES

WHERE PAYMENTNO <> PEN.PAYMENTNO)

Example 8.100:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE 1 <> ALL

(SELECT TEAMNO

FROM MATCHES

WHERE PLAYERNO = PLAYERS.PLAYERNO)

Example 8.101:

SELECT TEAMNO

FROM TEAMS

WHERE NOT EXISTS

(SELECT \*

FROM MATCHES

WHERE PLAYERNO = 57

AND TEAMNO = TEAMS.TEAMNO)

Example 8.102:

SELECT PLAYERNO

FROM PLAYERS AS P

WHERE NOT EXISTS

(SELECT \*

FROM TEAMS AS T

WHERE NOT EXISTS

(SELECT \*

FROM MATCHES AS M

WHERE T.TEAMNO = M.TEAMNO

AND P.PLAYERNO = M.PLAYERNO))

;

SELECT \*

FROM TEAMS AS T

WHERE NOT EXISTS

(SELECT \*

FROM MATCHES AS M

WHERE T.TEAMNO = M.TEAMNO

AND M.PLAYERNO = 27)

Example 8.103:

SELECT PLAYERNO

FROM PLAYERS

WHERE NOT EXISTS

(SELECT \*

FROM MATCHES AS M1

WHERE PLAYERNO = 57

AND NOT EXISTS

(SELECT \*

FROM MATCHES AS M2

WHERE M1.TEAMNO = M2.TEAMNO

AND PLAYERS.PLAYERNO = M2.PLAYERNO))

Example 8.104:

SELECT PLAYERNO

FROM MATCHES

WHERE TEAMNO IN

(SELECT TEAMNO

FROM TEAMS

WHERE TEAMNO NOT IN

(SELECT TEAMNO

FROM MATCHES

WHERE PLAYERNO = 57))

;

SELECT PLAYERNO

FROM PLAYERS AS P

WHERE NOT EXISTS

(SELECT \*

FROM MATCHES AS M1

WHERE PLAYERNO = 57

AND NOT EXISTS

(SELECT \*

FROM MATCHES AS M2

WHERE M1.TEAMNO = M2.TEAMNO

AND P.PLAYERNO = M2.PLAYERNO))

AND PLAYERNO NOT IN

(SELECT PLAYERNO

FROM MATCHES

WHERE TEAMNO IN

(SELECT TEAMNO

FROM TEAMS

WHERE TEAMNO NOT IN

(SELECT TEAMNO

FROM MATCHES

WHERE PLAYERNO = 57)))

Example 8.105:

SELECT PLAYERNO

FROM PLAYERS

WHERE TOWN = 'Stratford'

;

SELECT PLAYERNO

FROM PLAYERS

WHERE NOT (TOWN = 'Stratford')

;

SELECT PLAYERNO

FROM PLAYERS

WHERE TOWN <> 'Stratford'

Example 8.106:

SELECT PLAYERNO

FROM PENALTIES

WHERE AMOUNT = 25

;

SELECT PLAYERNO

FROM PENALTIES

WHERE AMOUNT <> 25

;

SELECT PLAYERNO

FROM PLAYERS

WHERE PLAYERNO NOT IN

(SELECT PLAYERNO

FROM PENALTIES

WHERE AMOUNT = 25)

Example 8.107:

SELECT PLAYERNO

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PENALTIES

WHERE AMOUNT = 25)

Answer 8.1:

SELECT PAYMENTNO

FROM PENALTIES

WHERE AMOUNT > 60

;

SELECT PAYMENTNO

FROM PENALTIES

WHERE 60 < AMOUNT

;

SELECT PAYMENTNO

FROM PENALTIES

WHERE AMOUNT - 60 > 0

Answer 8.5:

SELECT DISTINCT PLAYERNO

FROM MATCHES

WHERE WON + LOST = 5

Answer 8.6:

SELECT PLAYERNO, NAME, INITIALS

FROM PLAYERS

WHERE PLAYERNO =

(SELECT PLAYERNO

FROM PENALTIES

WHERE PAYMENTNO = 4)

Answer 8.7:

SELECT PLAYERNO, NAME, INITIALS

FROM PLAYERS

WHERE PLAYERNO =

(SELECT PLAYERNO

FROM TEAMS

WHERE TEAMNO =

(SELECT TEAMNO

FROM MATCHES

WHERE MATCHNO = 2))

Answer 8.8:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE BIRTH\_DATE =

(SELECT BIRTH\_DATE

FROM PLAYERS

WHERE NAME = 'Parmenter'

AND INITIALS = 'R')

AND NOT (NAME = 'Parmenter'

AND INITIALS = 'R')

Answer 8.9:

SELECT MATCHNO

FROM MATCHES

WHERE WON =

(SELECT WON

FROM MATCHES

WHERE MATCHNO = 6)

AND MATCHNO <> 6

AND TEAMNO = 2

Answer 8.10:

SELECT MATCHNO

FROM MATCHES

WHERE (WON, LOST) =

((SELECT WON

FROM MATCHES

WHERE MATCHNO = 2),

(SELECT LOST

FROM MATCHES

WHERE MATCHNO = 8))

Answer 8.11:

SELECT PLAYERNO, TOWN, STREET, HOUSENO

FROM PLAYERS

WHERE (TOWN, STREET, HOUSENO) <

(SELECT TOWN, STREET, HOUSENO

FROM PLAYERS

WHERE PLAYERNO = 100)

ORDER BY TOWN, STREET, HOUSENO

Answer 8.12:

SELECT PAYMENTNO

FROM PENALTIES

WHERE 1965 <

(SELECT YEAR(BIRTH\_DATE)

FROM PLAYERS

WHERE PLAYERS.PLAYERNO = PENALTIES.PLAYERNO)

Answer 8.13:

SELECT PAYMENTNO, PLAYERNO

FROM PENALTIES

WHERE PLAYERNO =

(SELECT PLAYERNO

FROM TEAMS

WHERE TEAMS.PLAYERNO = PENALTIES.PLAYERNO)

Answer 8.14:

SELECT PLAYERNO, NAME, TOWN

FROM PLAYERS

WHERE SEX = 'F'

AND TOWN <> 'Stratford'

;

SELECT PLAYERNO, NAME, TOWN

FROM PLAYERS

WHERE SEX = 'F'

AND NOT (TOWN = 'Stratford')

Answer 8.15:

SELECT PLAYERNO

FROM PLAYERS

WHERE JOINED >= 1970

AND JOINED <= 1980

;

SELECT PLAYERNO

FROM PLAYERS

WHERE NOT (JOINED < 1970 OR JOINED > 1980)

Answer 8.16:

SELECT PLAYERNO, NAME, BIRTH\_DATE

FROM PLAYERS

WHERE MOD(YEAR(BIRTH\_DATE), 400) = 0

OR (MOD(YEAR(BIRTH\_DATE), 4) = 0

AND NOT(MOD(YEAR(BIRTH\_DATE), 100) = 0))

Answer 8.17:

SELECT MATCHNO, NAME, INITIALS, DIVISION

FROM MATCHES AS M, PLAYERS AS P, TEAMS AS T

WHERE M.PLAYERNO = P.PLAYERNO

AND M.TEAMNO = T.TEAMNO

AND YEAR(BIRTH\_DATE) > 1965

AND WON > LOST

Answer 8.18:

SELECT PAYMENTNO

FROM PENALTIES

WHERE AMOUNT IN (50, 75, 100)

Answer 8.19:

SELECT PLAYERNO

FROM PLAYERS

WHERE TOWN NOT IN ('Stratford', 'Douglas')

;

SELECT PLAYERNO

FROM PLAYERS

WHERE NOT (TOWN IN ('Stratford', 'Douglas'))

;

SELECT PLAYERNO

FROM PLAYERS

WHERE TOWN <> 'Stratford'

AND TOWN <> 'Douglas'

Answer 8.20:

SELECT PAYMENTNO

FROM PENALTIES

WHERE AMOUNT IN

(100, PAYMENTNO \* 5,

(SELECT AMOUNT

FROM PENALTIES

WHERE PAYMENTNO = 2))

Answer 8.21:

SELECT PLAYERNO, TOWN, STREET

FROM PLAYERS

WHERE (TOWN, STREET) IN

(('Stratford','Haseltine Lane'),

('Stratford','Edgecombe Way'))

Answer 8.22:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PENALTIES)

Answer 8.23:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PENALTIES

WHERE AMOUNT > 50)

Answer 8.24:

SELECT TEAMNO, PLAYERNO

FROM TEAMS

WHERE DIVISION = 'first'

AND PLAYERNO IN

(SELECT PLAYERNO

FROM PLAYERS

WHERE TOWN = 'Stratford')

Answer 8.25:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PENALTIES)

AND PLAYERNO NOT IN

(SELECT PLAYERNO

FROM TEAMS

WHERE DIVISION = 'first')

;

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PENALTIES

WHERE PLAYERNO NOT IN

(SELECT PLAYERNO

FROM TEAMS

WHERE DIVISION = 'first'))

Answer 8.27:

SELECT MATCHNO, PLAYERNO

FROM MATCHES

WHERE (WON, LOST) IN

(SELECT WON, LOST

FROM MATCHES

WHERE TEAMNO IN

(SELECT TEAMNO

FROM TEAMS

WHERE DIVISION = 'second'))

Answer 8.28:

SELECT PLAYERNO, NAME

FROM PLAYERS AS P1

WHERE (TOWN, STREET, HOUSENO, POSTCODE) IN

(SELECT TOWN, STREET, HOUSENO, POSTCODE

FROM PLAYERS AS P2

WHERE P1.PLAYERNO <> P2.PLAYERNO)

Answer 8.29:

SELECT PAYMENTNO

FROM PENALTIES

WHERE AMOUNT BETWEEN 50 AND 100

Answer 8.30:

SELECT PAYMENTNO

FROM PENALTIES

WHERE NOT (AMOUNT BETWEEN 50 AND 100)

;

SELECT PAYMENTNO

FROM PENALTIES

WHERE AMOUNT NOT BETWEEN 50 AND 100

;

SELECT PAYMENTNO

FROM PENALTIES

WHERE AMOUNT < 50

OR AMOUNT > 100

Answer 8.31:

SELECT PLAYERNO

FROM PLAYERS

WHERE JOINED BETWEEN

YEAR(BIRTH\_DATE + INTERVAL 16 YEAR + INTERVAL 1 DAY)

AND YEAR(BIRTH\_DATE + INTERVAL 40 YEAR – INTERVAL 1 DAY)

Answer 8.32:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE NAME LIKE '%is%'

Answer 8.33:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE NAME LIKE '\_\_\_\_\_\_'

Answer 8.34:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE NAME LIKE '\_\_\_\_\_\_%'

;

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE NAME LIKE '%\_\_\_\_\_\_'

;

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE NAME LIKE '%\_\_\_\_\_\_%'

;

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE LENGTH(RTRIM(NAME)) > 6

Answer 8.35:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE NAME LIKE '\_r%r\_'

Answer 8.36:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE TOWN LIKE '\_@%%@%\_' ESCAPE '@'

Answer 8.37:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE NAME REGEXP 'en'

Answer 8.38:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE NAME REGEXP '^n.\*e$'

Answer 8.39:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE NAME REGEXP '[a-z]{9}'

Answer 8.40:

SELECT BOOKNO, SUMMARY

FROM BOOKS

WHERE MATCH(SUMMARY)

AGAINST ('students' IN NATURAL LANGUAGE MODE)

Answer 8.41:

SELECT BOOKNO, SUMMARY

FROM BOOKS

WHERE MATCH(SUMMARY)

AGAINST ('database' IN BOOLEAN MODE)

Answer 8.42:

SELECT BOOKNO, SUMMARY

FROM BOOKS

WHERE MATCH(SUMMARY)

AGAINST ('database languages'

IN NATURAL LANGUAGE MODE)

Answer 8.43:

SELECT BOOKNO, SUMMARY

FROM BOOKS

WHERE MATCH(SUMMARY)

AGAINST ('+database -languages' IN BOOLEAN MODE)

Answer 8.44:

SELECT PLAYERNO

FROM PLAYERS

WHERE LEAGUENO IS NULL

Answer 8.46:

SELECT NAME, INITIALS

FROM PLAYERS

WHERE EXISTS

(SELECT \*

FROM TEAMS

WHERE PLAYERNO = PLAYERS.PLAYERNO)

Answer 8.47:

SELECT NAME, INITIALS

FROM PLAYERS AS P

WHERE NOT EXISTS

(SELECT \*

FROM TEAMS AS T

WHERE T.PLAYERNO = P.PLAYERNO

AND EXISTS

(SELECT \*

FROM MATCHES AS M

WHERE M.TEAMNO = T.TEAMNO

AND M.PLAYERNO = 112))

Answer 8.48:

SELECT PLAYERNO

FROM PLAYERS

WHERE BIRTH\_DATE <= ALL

(SELECT BIRTH\_DATE

FROM PLAYERS

WHERE TOWN = 'Stratford')

AND TOWN = 'Stratford'

Answer 8.49:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE PLAYERNO = ANY

(SELECT PLAYERNO

FROM PENALTIES)

Answer 8.50:

SELECT PAYMENTNO, AMOUNT, PAYMENT\_DATE

FROM PENALTIES AS PEN1

WHERE AMOUNT >= ALL

(SELECT AMOUNT

FROM PENALTIES AS PEN2

WHERE YEAR(PEN1.PAYMENT\_DATE) =

YEAR(PEN2.PAYMENT\_DATE))

Answer 8.51:

SELECT (SELECT PLAYERNO

FROM PLAYERS

WHERE PLAYERNO <= ALL

(SELECT PLAYERNO

FROM PLAYERS)),

(SELECT PLAYERNO

FROM PLAYERS

WHERE PLAYERNO >= ALL

(SELECT PLAYERNO

FROM PLAYERS))

Answer 8.53:

SELECT NAME, INITIALS

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM MATCHES

WHERE TEAMNO IN

(SELECT TEAMNO

FROM TEAMS

WHERE DIVISION = 'first'))

AND PLAYERNO IN

(SELECT PLAYERNO

FROM MATCHES

WHERE WON > LOST)

AND PLAYERNO NOT IN

(SELECT PLAYERNO

FROM PENALTIES)

Answer 8.54:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM MATCHES

WHERE TEAMNO = 1)

AND PLAYERNO IN

(SELECT PLAYERNO

FROM MATCHES

WHERE TEAMNO = 2)

Answer 8.55:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE EXISTS

(SELECT \*

FROM PENALTIES

WHERE PLAYERNO = PLAYERS.PLAYERNO)

Answer 8.56:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM MATCHES AS M1

WHERE WON > LOST

AND EXISTS

(SELECT \*

FROM MATCHES AS M2

WHERE M1.PLAYERNO = M2.PLAYERNO

AND WON > LOST

AND M1.MATCHNO <> M2.MATCHNO))

;

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE 1 < (SELECT COUNT(\*)

FROM MATCHES

WHERE WON > LOST

AND PLAYERS.PLAYERNO = PLAYERNO)

Answer 8.57:

SELECT NAME, INITIALS

FROM PLAYERS

WHERE NOT EXISTS

(SELECT \*

FROM PENALTIES

WHERE PLAYERS.PLAYERNO = PLAYERNO

AND PAYMENT\_DATE BETWEEN '1980-01-01'

AND '1980-12-31')

Answer 8.58:

SELECT DISTINCT PLAYERNO

FROM PENALTIES AS PEN1

WHERE EXISTS

(SELECT \*

FROM PENALTIES AS PEN2

WHERE PEN1.AMOUNT = PEN2.AMOUNT

AND PEN1.PAYMENTNO <> PEN2.PAYMENTNO)

Answer 8.59:

SELECT PLAYERNO

FROM PLAYERS

WHERE PLAYERNO NOT IN

(SELECT PLAYERNO

FROM MATCHES WHERE WON = 3)

Answer 8.60:

SELECT TEAMNO, DIVISION

FROM TEAMS

WHERE TEAMNO NOT IN

(SELECT TEAMNO

FROM MATCHES

WHERE PLAYERNO = 6)

Answer 8.61:

SELECT DISTINCT PLAYERNO

FROM MATCHES

WHERE PLAYERNO NOT IN

(SELECT PLAYERNO

FROM MATCHES

WHERE TEAMNO IN

(SELECT TEAMNO

FROM MATCHES

WHERE PLAYERNO = 57))

Example 9.1:

SELECT \*

FROM PENALTIES

;

SELECT PAYMENTNO, PLAYERNO, PAYMENT\_DATE, AMOUNT

FROM PENALTIES

Example 9.2:

SELECT PENALTIES.\*

FROM PENALTIES INNER JOIN TEAMS

ON PENALTIES.PLAYERNO = TEAMS.PLAYERNO

;

SELECT PENALTIES.PAYMENTNO, PENALTIES.PLAYERNO,

PENALTIES.PAYMENT\_DATE, PENALTIES.AMOUNT

FROM PENALTIES INNER JOIN TEAMS

ON PENALTIES.PLAYERNO = TEAMS.PLAYERNO

;

SELECT PEN.\*

FROM PENALTIES AS PEN INNER JOIN TEAMS

ON PEN.PLAYERNO = TEAMS.PLAYERNO

Example 9.3:

SELECT MATCHNO, 'Tally', WON - LOST,

WON \* 10

FROM MATCHES

Example 9.4:

SELECT TOWN

FROM PLAYERS

;

SELECT DISTINCT TOWN

FROM PLAYERS

Example 9.5:

SELECT STREET, TOWN

FROM PLAYERS

;

SELECT DISTINCT STREET, TOWN

FROM PLAYERS

;

SELECT TOWN

FROM PLAYERS

;

SELECT ALL TOWN

FROM PLAYERS

Example 9.6:

SELECT DISTINCT LEAGUENO

FROM PLAYERS

Example 9.7:

SELECT DISTINCT \*

FROM (SELECT 1 AS A, 'Hello' AS B, 4 AS C UNION

SELECT 1, 'Hello', NULL UNION

SELECT 1, 'Hello', NULL UNION

SELECT 1, NULL, NULL) AS X

Example 9.8:

SELECT COUNT(\*)

FROM PLAYERS

Example 9.9:

SELECT COUNT(\*)

FROM PLAYERS

WHERE TOWN = 'Stratford'

;

SELECT COUNT(\*), PLAYERNO

FROM PLAYERS

;

SELECT 'The number of players', COUNT(\*)

FROM PLAYERS

Example 9.10:

SELECT COUNT(LEAGUENO)

FROM PLAYERS

;

SELECT COUNT(ALL LEAGUENO)

FROM PLAYERS

Example 9.11:

SELECT COUNT(DISTINCT TOWN)

FROM PLAYERS

Example 9.12:

SELECT COUNT(DISTINCT SUBSTR(NAME, 1, 1))

FROM PLAYERS

Example 9.13:

SELECT COUNT(DISTINCT YEAR(PAYMENT\_DATE))

FROM PENALTIES

Example 9.14:

SELECT COUNT(DISTINCT TOWN), COUNT(DISTINCT SEX)

FROM PLAYERS

Example 9.15:

SELECT PLAYERNO, NAME

FROM PLAYERS AS P

WHERE (SELECT COUNT(\*)

FROM PENALTIES AS PEN

WHERE P.PLAYERNO = PEN.PLAYERNO)

>

(SELECT COUNT(\*)

FROM MATCHES AS M

WHERE P.PLAYERNO = M.PLAYERNO)

Example 9.16:

SELECT PLAYERNO, NAME,

(SELECT COUNT(\*)

FROM PENALTIES

WHERE PENALTIES.PLAYERNO = PLAYERS.PLAYERNO)

AS NUMBER

FROM PLAYERS

WHERE (SELECT COUNT(\*)

FROM PENALTIES

WHERE PENALTIES.PLAYERNO = PLAYERS.PLAYERNO) >= 2

;

SELECT PLAYERNO, NAME, NUMBER

FROM (SELECT PLAYERNO, NAME,

(SELECT COUNT(\*)

FROM PENALTIES

WHERE PENALTIES.PLAYERNO =

PLAYERS.PLAYERNO)

AS NUMBER

FROM PLAYERS) AS PN

WHERE NUMBER >= 2

Example 9.17:

SELECT (SELECT COUNT(\*)

FROM PENALTIES),

(SELECT COUNT(\*)

FROM MATCHES)

Example 9.18:

SELECT MAX(AMOUNT)

FROM PENALTIES

Example 9.19:

SELECT MIN(AMOUNT)

FROM PENALTIES

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PLAYERS

WHERE TOWN = 'Stratford')

Example 9.20:

SELECT COUNT(\*)

FROM PENALTIES

WHERE AMOUNT =

(SELECT MIN(AMOUNT)

FROM PENALTIES)

Example 9.21:

SELECT DISTINCT TEAMNO, PLAYERNO

FROM MATCHES AS M1

WHERE WON =

(SELECT MAX(WON)

FROM MATCHES AS M2

WHERE M1.TEAMNO = M2.TEAMNO)

Example 9.22:

SELECT (MAX(AMOUNT) - MIN(AMOUNT)) \* 100

FROM PENALTIES

Example 9.23:

SELECT SUBSTR(MAX(NAME), 1, 1)

FROM PLAYERS

Example 9.24:

SELECT MAX(LEAGUENO)

FROM PLAYERS

WHERE TOWN = 'Midhurst'

Example 9.25:

SELECT CASE WHEN MIN(LEAGUENO) IS NULL

THEN 'Unknown'

ELSE MIN(LEAGUENO)

END

FROM PLAYERS

WHERE TOWN = 'Amsterdam'

Example 9.26:

SELECT PLAYERNO, AMOUNT, PAYMENT\_DATE

FROM PENALTIES AS PEN1

WHERE AMOUNT =

(SELECT MAX(AMOUNT)

FROM PENALTIES AS PEN2

WHERE PEN2.PLAYERNO = PEN1.PLAYERNO)

Example 9.27:

SELECT PLAYERNO,

(SELECT MAX(AMOUNT)

FROM PENALTIES

WHERE PENALTIES.PLAYERNO = PLAYERS.PLAYERNO)

AS HIGHESTPENALTY,

(SELECT MAX(WON)

FROM MATCHES

WHERE MATCHES.PLAYERNO = PLAYERS.PLAYERNO)

AS NUMBEROFSETS

FROM PLAYERS

Example 9.28:

SELECT PLAYERNO

FROM PLAYERS

WHERE (SELECT MIN(AMOUNT)

FROM PENALTIES

WHERE PENALTIES.PLAYERNO = PLAYERS.PLAYERNO) =

(SELECT MAX(AMOUNT)

FROM PENALTIES

WHERE PENALTIES.PLAYERNO = PLAYERS.PLAYERNO)

Example 9.29:

SELECT SUM(AMOUNT)

FROM PENALTIES

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PLAYERS

WHERE TOWN = 'Inglewood')

;

SELECT SUM(DISTINCT AMOUNT)

FROM PENALTIES

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PLAYERS

WHERE TOWN = 'Inglewood')

Example 9.30:

SELECT AVG(AMOUNT)

FROM PENALTIES

WHERE PLAYERNO = 44

Example 9.31:

SELECT DISTINCT PLAYERNO

FROM PENALTIES

WHERE AMOUNT >

(SELECT AVG(AMOUNT)

FROM PENALTIES)

Example 9.32:

SELECT AVG(DISTINCT AMOUNT)

FROM PENALTIES

Example 9.33:

SELECT AVG(LENGTH(RTRIM(NAME))), MAX(LENGTH(RTRIM(NAME)))

FROM PLAYERS

Example 9.34:

SELECT PAYMENTNO, AMOUNT,

ABS(AMOUNT - (SELECT AVG(AMOUNT)

FROM PENALTIES)) AS DIFFERENCE

FROM PENALTIES AS P

Example 9.35:

SELECT VARIANCE(AMOUNT)

FROM PENALTIES

WHERE PLAYERNO = 44

;

SELECT AMOUNT –

(SELECT AVG(AMOUNT)

FROM PENALTIES

WHERE PLAYERNO = 44)

FROM PENALTIES

WHERE PLAYERNO = 44

SELECT SUM(P)

FROM (SELECT POWER(AMOUNT -

(SELECT AVG(AMOUNT)

FROM PENALTIES

WHERE PLAYERNO = 44),2) AS P

FROM PENALTIES

WHERE PLAYERNO = 44) AS POWERS

;

SELECT SUM(P) /

(SELECT COUNT(\*) FROM PENALTIES WHERE PLAYERNO = 44)

FROM (SELECT POWER(AMOUNT -

(SELECT AVG(AMOUNT)

FROM PENALTIES

WHERE PLAYERNO = 44),2) AS P

FROM PENALTIES

WHERE PLAYERNO = 44) AS POWERS

Example 9.36:

SELECT STDDEV(AMOUNT)

FROM PENALTIES

WHERE PLAYERNO = 44

Example 9.37:

SELECT VAR\_SAMP(AMOUNT), VARIANCE(AMOUNT)

FROM PENALTIES

Example 9.38:

SELECT STDDEV\_SAMP(AMOUNT), STDDEV(AMOUNT)

FROM PENALTIES

Example 9.39:

CREATE TABLE BITS

(BIN\_VALUE INTEGER NOT NULL PRIMARY KEY)

;

INSERT INTO BITS

VALUES (CONV(001,2,16)),

(CONV(011,2,16)),

(CONV(111,2,16))

Example 9.40:

SELECT BIN(BIT\_OR(BIN\_VALUE))

FROM BITS

Answer 9.4:

SELECT COUNT(\*), MAX(AMOUNT)

FROM PENALTIES

Answer 9.5:

SELECT COUNT(DISTINCT POSITION)

FROM COMMITTEE\_MEMBERS

Answer 9.6:

SELECT COUNT(LEAGUENO)

FROM PLAYERS

WHERE TOWN = 'Inglewood'

Answer 9.7:

SELECT TEAMNO, DIVISION,

(SELECT COUNT(\*)

FROM MATCHES

WHERE TEAMS.TEAMNO = MATCHES.TEAMNO)

FROM TEAMS

Answer 9.8:

SELECT PLAYERNO, NAME,

(SELECT COUNT(\*)

FROM MATCHES

WHERE MATCHES.PLAYERNO = PLAYERS.PLAYERNO

AND WON > LOST)

FROM PLAYERS

Answer 9.9:

SELECT 'Number of players' AS TABLES,

(SELECT COUNT(\*) FROM PLAYERS) AS NUMBERS UNION

SELECT 'Number of teams',

(SELECT COUNT(\*) FROM TEAMS) UNION

SELECT 'Number of matches',

(SELECT COUNT(\*) FROM MATCHES)

Answer 9.10:

SELECT MIN(WON)

FROM MATCHES

WHERE WON > LOST

Answer 9.11:

SELECT PLAYERNO,

(SELECT MAX(AMOUNT)

FROM PENALTIES

WHERE PENALTIES.PLAYERNO =

PLAYERS.PLAYERNO) -

(SELECT MIN(AMOUNT)

FROM PENALTIES

WHERE PENALTIES.PLAYERNO =

PLAYERS.PLAYERNO)

FROM PLAYERS

Answer 9.12:

SELECT PLAYERNO, BIRTH\_DATE

FROM PLAYERS

WHERE YEAR(BIRTH\_DATE) =

(SELECT MAX(YEAR(BIRTH\_DATE))

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM MATCHES

WHERE TEAMNO = 1))

Answer 9.14:

SELECT AVG(AMOUNT)

FROM PENALTIES

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM MATCHES

WHERE TEAMNO = 1)

Answer 9.15:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE (SELECT SUM(AMOUNT)

FROM PENALTIES

WHERE PENALTIES.PLAYERNO = PLAYERS.PLAYERNO)

> 100

Answer 9.16:

SELECT NAME, INITIALS

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM MATCHES

WHERE WON >

(SELECT SUM(WON)

FROM MATCHES

WHERE PLAYERNO = 27))

Answer 9.17:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE (SELECT SUM(WON)

FROM MATCHES

WHERE MATCHES.PLAYERNO =

PLAYERS.PLAYERNO) = 8

Answer 9.18:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE LENGTH(RTRIM(NAME)) >

(SELECT AVG(LENGTH(RTRIM(NAME)))

FROM PLAYERS)

Answer 9.19:

SELECT PLAYERNO,

(SELECT MAX(AMOUNT)

FROM PENALTIES

WHERE PENALTIES.PLAYERNO =

PLAYERS.PLAYERNO) -

(SELECT AVG(AMOUNT)

FROM PENALTIES

WHERE PENALTIES.PLAYERNO =

PLAYERS.PLAYERNO)

FROM PLAYERS

Answer 9.20:

SELECT PLAYERNO,

REPEAT('\*',

CAST((SELECT AVG(AMOUNT)

FROM PENALTIES

WHERE PENALTIES.PLAYERNO =

PLAYERS.PLAYERNO)/10

AS SIGNED INTEGER))

FROM PLAYERS

Answer 9.21:

SELECT SQRT(SUM(P) /

(SELECT COUNT(\*) FROM PENALTIES WHERE

PLAYERNO = 44))

FROM (SELECT POWER(AMOUNT -

(SELECT AVG(AMOUNT)

FROM PENALTIES

WHERE PLAYERNO = 44),2) AS P

FROM PENALTIES

WHERE PLAYERNO = 44) AS POWERS

Example 10.1:

SELECT TOWN

FROM PLAYERS

GROUP BY TOWN

Example 10.2:

SELECT TOWN, COUNT(\*)

FROM PLAYERS

GROUP BY TOWN

Example 10.3:

SELECT TEAMNO, COUNT(\*), SUM(WON)

FROM MATCHES

GROUP BY TEAMNO

Example 10.4:

SELECT TEAMNO, COUNT(\*)

FROM MATCHES

WHERE TEAMNO IN

(SELECT TEAMNO

FROM TEAMS INNER JOIN PLAYERS

ON TEAMS.PLAYERNO = PLAYERS.PLAYERNO

WHERE TOWN = 'Eltham')

GROUP BY TEAMNO

Example 10.5:

SELECT AMOUNT, COUNT(\*), SUM(AMOUNT)

FROM PENALTIES

GROUP BY AMOUNT

Example 10.6:

SELECT TEAMNO, PLAYERNO

FROM MATCHES

GROUP BY TEAMNO, PLAYERNO

;

SELECT TEAMNO, PLAYERNO

FROM MATCHES

GROUP BY PLAYERNO, TEAMNO

;

SELECT TEAMNO, PLAYERNO, SUM(WON),

COUNT(\*), MIN(LOST)

FROM MATCHES

GROUP BY TEAMNO, PLAYERNO

Example 10.7:

SELECT P.PLAYERNO, NAME, SUM(AMOUNT)

FROM PLAYERS AS P INNER JOIN PENALTIES AS PEN

ON P.PLAYERNO = PEN.PLAYERNO

GROUP BY P.PLAYERNO, NAME

Example 10.8:

SELECT YEAR(PAYMENT\_DATE), COUNT(\*)

FROM PENALTIES

GROUP BY YEAR(PAYMENT\_DATE)

Example 10.9:

SELECT TRUNCATE(PLAYERNO/25,0), COUNT(\*), MAX(PLAYERNO)

FROM PLAYERS

GROUP BY TRUNCATE(PLAYERNO/25,0)

Example 10.10:

SELECT LEAGUENO

FROM PLAYERS

GROUP BY LEAGUENO

Example 10.11:

SELECT TEAMNO, COUNT(\*)

FROM MATCHES

GROUP BY TEAMNO

ORDER BY TEAMNO DESC

;

SELECT TEAMNO, COUNT(\*)

FROM MATCHES

GROUP BY TEAMNO DESC

Example 10.12:

SELECT CAST(AMOUNT \* 100 AS SIGNED INTEGER)

AS AMOUNT\_IN\_CENTS

FROM PENALTIES

GROUP BY AMOUNT

Example 10.13:

SELECT TEAMNO, GROUP\_CONCAT(PLAYERNO)

FROM MATCHES

GROUP BY TEAMNO

Example 10.14:

SELECT TEAMNO, GROUP\_CONCAT(TEAMNO)

FROM MATCHES

GROUP BY TEAMNO

Example 10.15:

SELECT GROUP\_CONCAT(PAYMENTNO)

FROM PENALTIES

Example 10.16:

SET @@GROUP\_CONCAT\_MAX\_LEN=7

;

SELECT TEAMNO, GROUP\_CONCAT(TEAMNO)

FROM MATCHES

GROUP BY TEAMNO

Example 10.17:

SELECT AVG(TOTAL)

FROM (SELECT PLAYERNO, SUM(AMOUNT) AS TOTAL

FROM PENALTIES

GROUP BY PLAYERNO) AS TOTALS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PLAYERS

WHERE TOWN = 'Stratford' OR TOWN = 'Inglewood')

Example 10.18:

SELECT PLAYERS.PLAYERNO, NAME, NUMBER\_OF\_PENALTIES,

NUMBER\_OF\_TEAMS

FROM PLAYERS,

(SELECT PLAYERNO, COUNT(\*) AS NUMBER\_OF\_PENALTIES

FROM PENALTIES

GROUP BY PLAYERNO) AS NUMBER\_PENALTIES,

(SELECT PLAYERNO, COUNT(\*) AS NUMBER\_OF\_TEAMS

FROM TEAMS

GROUP BY PLAYERNO) AS NUMBER\_TEAMS

WHERE PLAYERS.PLAYERNO = NUMBER\_PENALTIES.PLAYERNO

AND PLAYERS.PLAYERNO = NUMBER\_TEAMS.PLAYERNO

;

SELECT PLAYERS.PLAYERNO, NAME,

(SELECT COUNT(\*)

FROM PENALTIES

WHERE PLAYERS.PLAYERNO =

PENALTIES.PLAYERNO) AS NUMBER\_OF\_PENALTIES,

(SELECT COUNT(\*)

FROM TEAMS

WHERE PLAYERS.PLAYERNO =

TEAMS.PLAYERNO) AS NUMBER\_OF\_TEAMS

FROM PLAYERS

Example 10.19:

SELECT DISTINCT M.PLAYERNO, NUMBERP

FROM MATCHES AS M LEFT OUTER JOIN

(SELECT PLAYERNO, COUNT(\*) AS NUMBERP

FROM PENALTIES

GROUP BY PLAYERNO) AS NP

ON M.PLAYERNO = NP.PLAYERNO

Example 10.20:

SELECT GROUPS.PGROUP, SUM(P.AMOUNT)

FROM PENALTIES AS P,

(SELECT 1 AS PGROUP, '1980-01-01' AS START,

'1981-06-30' AS END

UNION

SELECT 2, '1981-07-01', '1982-12-31'

UNION

SELECT 3, '1983-01-01', '1984-12-31') AS GROUPS

WHERE P.PAYMENT\_DATE BETWEEN START AND END

GROUP BY GROUPS.PGROUP

ORDER BY GROUPS.PGROUP

Example 10.21:

SELECT P1.PAYMENTNO, P1.AMOUNT, SUM(P2.AMOUNT)

FROM PENALTIES AS P1, PENALTIES AS P2

WHERE P1.PAYMENTNO >= P2. PAYMENTNO

GROUP BY P1. PAYMENTNO, P1.AMOUNT

ORDER BY P1. PAYMENTNO

Example 10.22:

SELECT P1.PAYMENTNO, P1.AMOUNT,

(P1.AMOUNT \* 100) / SUM(P2.AMOUNT)

FROM PENALTIES AS P1, PENALTIES AS P2

GROUP BY P1.PAYMENTNO, P1.AMOUNT

ORDER BY P1.PAYMENTNO

Example 10.23:

SELECT PLAYERNO, SUM(AMOUNT)

FROM PENALTIES

GROUP BY PLAYERNO

UNION

SELECT NULL, SUM(AMOUNT)

FROM PENALTIES

;

SELECT PLAYERNO, SUM(AMOUNT)

FROM PENALTIES

GROUP BY PLAYERNO WITH ROLLUP

Example 10.24:

SELECT SEX, TOWN, COUNT(\*)

FROM PLAYERS

GROUP BY SEX, TOWN WITH ROLLUP

Answer 10.1:

SELECT JOINED

FROM PLAYERS

GROUP BY JOINED

Answer 10.2:

SELECT JOINED, COUNT(\*)

FROM PLAYERS

GROUP BY JOINED

Answer 10.3:

SELECT PLAYERNO, AVG(AMOUNT), COUNT(\*)

FROM PENALTIES

GROUP BY PLAYERNO

Answer 10.4:

SELECT TEAMNO, COUNT(\*), SUM(WON)

FROM MATCHES

WHERE TEAMNO IN

(SELECT TEAMNO

FROM TEAMS

WHERE DIVISION = 'first')

GROUP BY TEAMNO

Answer 10.5:

SELECT WON, LOST, COUNT(\*)

FROM MATCHES

WHERE WON > LOST

GROUP BY WON, LOST

ORDER BY WON, LOST

Answer 10.6:

SELECT P.TOWN, T.DIVISION, SUM(WON)

FROM (MATCHES AS M INNER JOIN PLAYERS AS P

ON M.PLAYERNO = P.PLAYERNO)

INNER JOIN TEAMS AS T

ON M.TEAMNO = T.TEAMNO

GROUP BY P.TOWN, T.DIVISION

ORDER BY P.TOWN

Answer 10.7:

SELECT NAME, INITIALS, COUNT(\*)

FROM PLAYERS AS P INNER JOIN PENALTIES AS PEN

ON P.PLAYERNO = PEN.PLAYERNO

WHERE P.TOWN = 'Inglewood'

GROUP BY P.PLAYERNO, NAME, INITIALS

Answer 10.8:

SELECT T.TEAMNO, DIVISION, SUM(WON)

FROM TEAMS AS T, MATCHES AS M

WHERE T.TEAMNO = M.TEAMNO

GROUP BY T.TEAMNO, DIVISION

Answer 10.9:

SELECT LENGTH(RTRIM(NAME)), COUNT(\*)

FROM PLAYERS

GROUP BY LENGTH(RTRIM(NAME))

Answer 10.10:

SELECT ABS(WON - LOST), COUNT(\*)

FROM MATCHES

GROUP BY ABS(WON – LOST)

Answer 10.11:

SELECT YEAR(BEGIN\_DATE), MONTH(BEGIN\_DATE), COUNT(\*)

FROM COMMITTEE\_MEMBERS

GROUP BY YEAR(BEGIN\_DATE), MONTH(BEGIN\_DATE)

ORDER BY YEAR(BEGIN\_DATE), MONTH(BEGIN\_DATE)

Answer 10.14:

SELECT AVG(NUMBERS)

FROM (SELECT COUNT(\*) AS NUMBERS

FROM PLAYERS

GROUP BY TOWN) AS TOWNS

Answer 10.15:

SELECT TEAMS.TEAMNO, DIVISION, NUMBER\_PLAYERS

FROM TEAMS LEFT OUTER JOIN

(SELECT TEAMNO, COUNT(\*) AS NUMBER\_PLAYERS

FROM MATCHES

GROUP BY TEAMNO) AS M

ON (TEAMS.TEAMNO = M.TEAMNO)

Answer 10.16:

SELECT PLAYERS.PLAYERNO, NAME, SUM\_AMOUNT,

NUMBER\_TEAMS

FROM (PLAYERS LEFT OUTER JOIN

(SELECT PLAYERNO, SUM(AMOUNT) AS SUM\_AMOUNT

FROM PENALTIES

GROUP BY PLAYERNO) AS TOTALS

ON (PLAYERS.PLAYERNO = TOTALS.PLAYERNO))

LEFT OUTER JOIN

(SELECT PLAYERNO, COUNT(\*) AS NUMBER\_TEAMS

FROM TEAMS

WHERE DIVISION = 'first'

GROUP BY PLAYERNO) AS NUMBERS

ON (PLAYERS.PLAYERNO = NUMBERS.PLAYERNO)

Answer 10.17:

SELECT TEAMNO, COUNT(DISTINCT PLAYERNO)

FROM MATCHES

WHERE TEAMNO IN

(SELECT TEAMNO

FROM PLAYERS AS P INNER JOIN TEAMS AS T

ON P.PLAYERNO = T.PLAYERNO

AND TOWN = 'Stratford')

AND WON > LOST

GROUP BY TEAMNO

Answer 10.18:

SELECT PLAYERNO, NAME, JOINED - AVERAGE

FROM PLAYERS,

(SELECT AVG(JOINED) AS AVERAGE

FROM PLAYERS) AS T

Answer 10.19:

SELECT PLAYERNO, NAME, JOINED – AVERAGE

FROM PLAYERS,

(SELECT TOWN, AVG(JOINED) AS AVERAGE

FROM PLAYERS

GROUP BY TOWN) AS TOWNS

WHERE PLAYERS.TOWN = TOWNS.TOWN

Answer 10.20:

SELECT TEAMNO, COUNT(\*)

FROM MATCHES

GROUP BY TEAMNO WITH ROLLUP

Answer 10.21:

SELECT P.NAME, T.DIVISION, SUM(WON)

FROM (MATCHES AS M INNER JOIN PLAYERS AS P

ON M.PLAYERNO = P.PLAYERNO)

INNER JOIN TEAMS AS T

ON M.TEAMNO = T.TEAMNO

GROUP BY P.NAME, T.DIVISION WITH ROLLUP

Example 11.1:

SELECT PLAYERNO

FROM PENALTIES

GROUP BY PLAYERNO

HAVING COUNT(\*) > 1

Example 11.2:

SELECT PLAYERNO

FROM PENALTIES

GROUP BY PLAYERNO

HAVING MAX(YEAR(PAYMENT\_DATE)) = 1984

Example 11.3:

SELECT PLAYERNO, SUM(AMOUNT)

FROM PENALTIES

GROUP BY PLAYERNO

HAVING SUM(AMOUNT) > 150

Example 11.4:

SELECT PLAYERNO, SUM(AMOUNT)

FROM PENALTIES

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM TEAMS)

GROUP BY PLAYERNO

HAVING SUM(AMOUNT) > 80

Example 11.5:

SELECT PLAYERNO, SUM(AMOUNT)

FROM PENALTIES

GROUP BY PLAYERNO

HAVING SUM(AMOUNT) >= ALL

(SELECT SUM(AMOUNT)

FROM PENALTIES

GROUP BY PLAYERNO)

Example 11.6:

SELECT SUM(AMOUNT)

FROM PENALTIES

HAVING SUM(AMOUNT) >= 250

Example 11.7:

SELECT GROUP\_CONCAT(PLAYERNO) AS LIST

FROM MATCHES

HAVING TRUE

Answer 11.1:

SELECT TOWN

FROM PLAYERS

GROUP BY TOWN

HAVING COUNT(\*) > 4

Answer 11.2:

SELECT PLAYERNO

FROM PENALTIES

GROUP BY PLAYERNO

HAVING SUM(AMOUNT) > 150

Answer 11.3:

SELECT NAME, INITIALS, COUNT(\*)

FROM PLAYERS INNER JOIN PENALTIES

ON PLAYERS.PLAYERNO = PENALTIES.PLAYERNO

GROUP BY PLAYERS.PLAYERNO, NAME, INITIALS

HAVING COUNT(\*) > 1

Answer 11.4:

SELECT TEAMNO, COUNT(\*)

FROM MATCHES

GROUP BY TEAMNO

HAVING COUNT(\*) >= ALL

(SELECT COUNT(\*)

FROM MATCHES

GROUP BY TEAMNO)

Answer 11.5:

SELECT TEAMNO, DIVISION

FROM TEAMS

WHERE TEAMNO IN

(SELECT TEAMNO

FROM MATCHES

GROUP BY TEAMNO

HAVING COUNT(DISTINCT PLAYERNO) > 4)

Answer 11.6:

SELECT NAME, INITIALS

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PENALTIES

WHERE AMOUNT > 40

GROUP BY PLAYERNO

HAVING COUNT(\*) >= 2)

Answer 11.7:

SELECT NAME, INITIALS

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PENALTIES

GROUP BY PLAYERNO

HAVING SUM(AMOUNT) >= ALL

(SELECT SUM(AMOUNT)

FROM PENALTIES

GROUP BY PLAYERNO))

Answer 11.8:

SELECT PLAYERNO

FROM PENALTIES

WHERE PLAYERNO <> 104

GROUP BY PLAYERNO

HAVING SUM(AMOUNT) =

(SELECT SUM(AMOUNT) \* 2

FROM PENALTIES

WHERE PLAYERNO = 104)

Answer 11.9:

SELECT PLAYERNO

FROM PENALTIES

WHERE PLAYERNO <> 6

GROUP BY PLAYERNO

HAVING COUNT(\*) =

(SELECT COUNT(\*)

FROM PENALTIES

WHERE PLAYERNO = 6)

Answer 11.10:

SELECT P.PLAYERNO, P.NAME

FROM PLAYERS AS P, MATCHES AS M1

WHERE P.PLAYERNO = M1.PLAYERNO

GROUP BY P.PLAYERNO, P.NAME

HAVING SUM(WON) >

(SELECT SUM(LOST)

FROM MATCHES AS M2

WHERE M2.PLAYERNO = P.PLAYERNO

GROUP BY M2.PLAYERNO)

Example 12.1:

SELECT PAYMENTNO, PLAYERNO

FROM PENALTIES

ORDER BY PLAYERNO

Example 12.2:

SELECT PLAYERNO, AMOUNT

FROM PENALTIES

ORDER BY PLAYERNO, AMOUNT

Example 12.3:

SELECT AMOUNT

FROM PENALTIES

ORDER BY PLAYERNO, AMOUNT

Example 12.4:

SELECT NAME, INITIALS, PLAYERNO

FROM PLAYERS

ORDER BY SUBSTR(NAME, 1, 1)

Example 12.5:

SELECT PLAYERNO, AMOUNT

FROM PENALTIES

ORDER BY ABS(AMOUNT - (SELECT AVG(AMOUNT) FROM PENALTIES))

Example 12.6:

SELECT PLAYERNO, AMOUNT

FROM PENALTIES AS P1

ORDER BY (SELECT AVG(AMOUNT)

FROM PENALTIES AS P2

WHERE P1.PLAYERNO = P2.PLAYERNO)

Example 12.7:

SELECT PLAYERNO, SUM(AMOUNT)

FROM PENALTIES

GROUP BY PLAYERNO

ORDER BY 2

Example 12.8:

SELECT PLAYERNO, NAME,

(SELECT SUM(AMOUNT)

FROM PENALTIES AS PEN

WHERE PEN.PLAYERNO=P.PLAYERNO)

FROM PLAYERS AS P

ORDER BY 3

;

SELECT PLAYERNO, NAME,

(SELECT SUM(AMOUNT)

FROM PENALTIES AS PEN

WHERE PEN.PLAYERNO=P.PLAYERNO) AS TOTAL

FROM PLAYERS AS P

ORDER BY TOTAL

Example 12.9:

SELECT PLAYERNO, AMOUNT

FROM PENALTIES

ORDER BY PLAYERNO DESC, AMOUNT ASC

Example 12.10:

CREATE TABLE CODES

(CODE CHAR(4) NOT NULL)

;

INSERT INTO CODES VALUES ('abc')

;

INSERT INTO CODES VALUES ('ABC')

;

INSERT INTO CODES VALUES ('-abc')

;

INSERT INTO CODES VALUES ('a bc')

;

INSERT INTO CODES VALUES ('ab')

;

INSERT INTO CODES VALUES ('9abc')

;

SELECT \*

FROM CODES

ORDER BY CODE

Example 12.11:

SELECT DISTINCT LEAGUENO

FROM PLAYERS

ORDER BY LEAGUENO DESC

Answer 12.3:

SELECT PLAYERNO, TEAMNO, WON - LOST

FROM MATCHES

ORDER BY 3 ASC

Example 13.1:

SELECT MAX(PLAYERNO)

FROM PLAYERS

;

SELECT PLAYERNO, NAME

FROM PLAYERS AS P1

WHERE 4 >

(SELECT COUNT(\*)

FROM PLAYERS AS P2

WHERE P1.PLAYERNO < P2.PLAYERNO)

ORDER BY PLAYERNO DESC

;

SELECT PLAYERNO, NAME

FROM PLAYERS

ORDER BY PLAYERNO DESC

LIMIT 4

Example 13.2:

SELECT LEAGUENO, PLAYERNO, NAME

FROM PLAYERS

ORDER BY LEAGUENO ASC

LIMIT 5

Example 13.3:

SELECT PLAYERNO, COUNT(\*) AS NUMBER

FROM MATCHES

WHERE WON > LOST

GROUP BY PLAYERNO

ORDER BY NUMBER DESC

LIMIT 3

Example 13.4:

SELECT PLAYERNO, COUNT(\*) AS NUMBER

FROM MATCHES

WHERE WON > LOST

GROUP BY PLAYERNO

ORDER BY NUMBER DESC, PLAYERNO DESC

LIMIT 3

Example 13.5:

SELECT \*

FROM (SELECT PLAYERNO, COUNT(\*) AS NUMBER

FROM MATCHES

WHERE WON > LOST

GROUP BY PLAYERNO

ORDER BY NUMBER DESC, PLAYERNO DESC

LIMIT 3) AS T

ORDER BY 1

Example 13.6:

SELECT AVG(AMOUNT)

FROM (SELECT AMOUNT

FROM PENALTIES

ORDER BY AMOUNT

LIMIT 4) AS T

Example 13.7:

SELECT MIN(AMOUNT)

FROM (SELECT AMOUNT

FROM PENALTIES

ORDER BY AMOUNT DESC

LIMIT 3) AS T

Example 13.8:

SELECT DISTINCT AMOUNT

FROM PENALTIES

ORDER BY AMOUNT DESC

LIMIT 3

Example 13.9:

SELECT PLAYERNO

FROM (SELECT PLAYERNO

FROM PLAYERS

WHERE LEAGUENO IS NOT NULL

ORDER BY LEAGUENO DESC

LIMIT 6) AS T

ORDER BY PLAYERNO

LIMIT 3

Example 13.10:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM (SELECT PLAYERNO, SUM(AMOUNT) AS TOTAL

FROM PENALTIES

GROUP BY PLAYERNO

ORDER BY TOTAL DESC

LIMIT 3) AS T)

Example 13.11:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PENALTIES)

AND PLAYERNO NOT IN

(SELECT PLAYERNO

FROM PENALTIES

ORDER BY AMOUNT DESC

LIMIT 2)

AND PLAYERNO NOT IN

(SELECT PLAYERNO

FROM PENALTIES

ORDER BY AMOUNT ASC

LIMIT 2)

;

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PENALTIES

WHERE PLAYERNO NOT IN

(SELECT PLAYERNO

FROM PENALTIES

ORDER BY AMOUNT DESC

LIMIT 2)

AND PLAYERNO NOT IN

(SELECT PLAYERNO

FROM PENALTIES

ORDER BY AMOUNT ASC

LIMIT 2))

Example 13.12:

SELECT PLAYERNO, NAME

FROM PLAYERS

ORDER BY PLAYERNO ASC

LIMIT 5 OFFSET 3

Example 13.13:

SELECT SQL\_CALC\_FOUND\_ROWS PAYMENTNO

FROM PENALTIES

LIMIT 5

;

SELECT FOUND\_ROWS()

Answer 13.1:

SELECT PAYMENTNO, AMOUNT, PAYMENT\_DATE

FROM PENALTIES

ORDER BY AMOUNT DESC, PAYMENT\_DATE DESC

LIMIT 4

Answer 13.2:

(SELECT MATCHNO

FROM MATCHES

ORDER BY MATCHNO ASC

LIMIT 2)

UNION

(SELECT MATCHNO

FROM MATCHES

ORDER BY MATCHNO DESC

LIMIT 2)

Answer 13.3:

SELECT PLAYERNO, NAME

FROM (SELECT PLAYERNO, NAME

FROM PLAYERS

ORDER BY PLAYERNO ASC

LIMIT 10) AS S10

ORDER BY NAME DESC

LIMIT 5

Answer 13.4:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM (SELECT PLAYERNO, COUNT(\*) AS NUMBER

FROM MATCHES

WHERE WON > LOST

GROUP BY PLAYERNO) AS WINNERS

ORDER BY NUMBER DESC, PLAYERNO ASC

LIMIT 2)

Answer 13.5:

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PENALTIES.PLAYERNO

FROM PENALTIES INNER JOIN PLAYERS

ON PENALTIES.PLAYERNO = PLAYERS.PLAYERNO

ORDER BY AMOUNT DESC, NAME ASC

LIMIT 4)

Answer 13.6:

SELECT PAYMENTNO, AMOUNT

FROM PENALTIES

ORDER BY AMOUNT DESC

LIMIT 1 OFFSET 2

Example 14.1:

SELECT PLAYERNO, TOWN

FROM PLAYERS

WHERE TOWN = 'Inglewood'

UNION

SELECT PLAYERNO, TOWN

FROM PLAYERS

WHERE TOWN = 'Plymouth'

;

SELECT PLAYERNO, TOWN

FROM PLAYERS

WHERE TOWN = 'Inglewood'

OR TOWN = 'Plymouth'

Example 14.2:

SELECT BIRTH\_DATE AS DATES

FROM PLAYERS

UNION

SELECT PAYMENT\_DATE

FROM PENALTIES

Example 14.3:

SELECT PLAYERNO

FROM PENALTIES

UNION

SELECT PLAYERNO

FROM TEAMS

Example 14.4:

SELECT PLAYERNO

FROM PENALTIES

UNION

SELECT PLAYERNO

FROM TEAMS

UNION

SELECT PLAYERNO

FROM PLAYERS

WHERE TOWN = 'Stratford'

Example 14.5:

SELECT CAST(TEAMNO AS CHAR(4)) AS TEAMNO,

CAST(PLAYERNO AS CHAR(4)) AS PLAYERNO,

SUM(WON + LOST) AS TOTAL

FROM MATCHES

GROUP BY TEAMNO, PLAYERNO

UNION

SELECT CAST(TEAMNO AS CHAR(4)),

'subtotal',

SUM(WON + LOST)

FROM MATCHES

GROUP BY TEAMNO

UNION

SELECT 'total', 'total', SUM(WON + LOST)

FROM MATCHES

ORDER BY 1, 2

Example 14.6:

SELECT PLAYERNO

FROM PENALTIES

UNION ALL

SELECT PLAYERNO

FROM TEAMS

Answer 14.1:

SELECT PLAYERNO

FROM COMMITTEE\_MEMBERS

UNION

SELECT PLAYERNO

FROM PENALTIES

GROUP BY PLAYERNO

HAVING COUNT(\*) >= 2

Answer 14.2:

SELECT MAX(ADATE)

FROM (SELECT MAX(BIRTH\_DATE) AS ADATE

FROM PLAYERS

UNION

SELECT MAX(PAYMENT\_DATE) AS ADATE

FROM PENALTIES) AS TWODATES

Answer 14.5:

SELECT SUM(NUMBER)

FROM (SELECT COUNT(\*) AS NUMBER

FROM PLAYERS

UNION ALL

SELECT COUNT(\*) AS NUMBER

FROM TEAMS) AS NUMBERS

Answer 14.6:

SELECT POWER(DIGIT,2)

FROM (SELECT 0 AS DIGIT UNION SELECT 1 UNION

SELECT 2 UNION SELECT 3 UNION

SELECT 4 UNION SELECT 5 UNION

SELECT 6 UNION SELECT 7 UNION

SELECT 8 UNION SELECT 9) AS DIGITS1

UNION ALL

SELECT POWER(DIGIT,3)

FROM (SELECT 0 AS DIGIT UNION SELECT 1 UNION

SELECT 2 UNION SELECT 3 UNION

SELECT 4 UNION SELECT 5 UNION

SELECT 6 UNION SELECT 7 UNION

SELECT 8 UNION SELECT 9) AS DIGITS2

ORDER BY 1

Example 15.1:

SET @PI = 3.141592654

Example 15.2:

SELECT @PI

Example 15.3:

SELECT NAME, TOWN, POSTCODE

FROM PLAYERS

WHERE PLAYERNO < @PI

Example 15.4:

SET @ABC = 5, @DEF = 'Inglewood',

@GHI = DATE('2004-01-01')

Example 15.5:

SET @PI = CAST(22 AS BINARY)/7

Example 15.6:

SET @ANR = (SELECT PLAYERNO

FROM TEAMS

WHERE TEAMNO = 1)

Example 15.7:

SELECT @PLAYERNO := 7

Example 15.8:

SELECT @NAME := 'Johnson', @TOWN := 'Inglewood',

@POSTCODE := '1234AB'

Example 15.9:

SELECT @NAME := NAME, @TOWN := TOWN,

@POSTCODE := POSTCODE

FROM PLAYERS

WHERE PLAYERNO = 2

Example 15.10:

SELECT @PENALTIESTOTAL := SUM(AMOUNT),

@NUMBERPENALTIES := COUNT(\*)

FROM PENALTIES

Example 15.11:

SELECT @PLAYERNO := PLAYERNO

FROM PLAYERS

ORDER BY PLAYERNO DESC

;

SELECT @PLAYERNO

;

SELECT @PNR7 := 7

FROM PLAYERS

WHERE PLAYERNO < @PNR7

Example 15.12:

SET @CNO = (SELECT PLAYERNO

FROM TEAMS

WHERE TEAMNO = 1)

;

SELECT NAME

FROM PLAYERS

WHERE PLAYERNO = @CNO

Example 15.13:

SET @VAR = (((3/7) \* 100)/124)+3

;

SELECT \*

FROM PENALTIES

WHERE PAYMENTNO < @VAR

AND PLAYERNO > @VAR

Example 15.14:

CREATE TABLE VARIABLES

(VARNAME CHAR(30) NOT NULL PRIMARY KEY,

VARVALUE CHAR(30) NOT NULL)

;

SET @VAR1 = 100, @VAR2 = 'John'

;

INSERT INTO VARIABLES VALUES ('VAR1', @VAR1)

;

INSERT INTO VARIABLES VALUES ('VAR2', @VAR2)

;

SELECT @VAR1 := VARVALUE

FROM VARIABLES

WHERE VARNAME = 'VAR1'

;

SELECT @VAR2 := VARVALUE

FROM VARIABLES

WHERE VARNAME = 'VAR2'

;

SELECT @VAR1, @VAR2

Example 15.15:

DO CURRENT\_DATE + INTERVAL 2 YEAR

Answer 15.1:

SET @TODAY = CURRENT\_DATE

;

SELECT @TODAY := CURRENT\_DATE

Answer 15.2:

SELECT \*

FROM PENALTIES

WHERE PENALTIES\_DATE < @TODAY - INTERVAL 5 YEAR

Answer 15.3:

SELECT @VAR := SUM(AMOUNT)

FROM PENALTIES

Example 16.1:

HANDLER PENALTIES OPEN

;

HANDLER PENALTIES READ FIRST

;

HANDLER PENALTIES READ NEXT

;

HANDLER PENALTIES CLOSE

Example 16.2:

CREATE INDEX PENALTIES\_AMOUNT ON PENALTIES (AMOUNT)

Example 16.3:

HANDLER PENALTIES OPEN AS P

;

HANDLER P READ PENALTIES\_AMOUNT FIRST

;

HANDLER P READ PENALTIES\_AMOUNT NEXT

Example 16.4:

HANDLER PENALTIES OPEN AS P

;

HANDLER P READ PENALTIES\_AMOUNT FIRST WHERE PLAYERNO > 100

;

HANDLER P READ PENALTIES\_AMOUNT NEXT WHERE PLAYERNO > 100

Example 16.5:

HANDLER PENALTIES OPEN AS P

;

HANDLER P READ PENALTIES\_AMOUNT FIRST LIMIT 3

Example 16.6:

HANDLER PENALTIES OPEN AS P

;

HANDLER P READ PENALTIES\_AMOUNT = (30.00)

;

HANDLER P READ PENALTIES\_AMOUNT NEXT

Example 16.7:

CREATE INDEX AMOUNT\_PLAYERNO ON PENALTIES (AMOUNT, PLAYERNO)

Example 16.8:

HANDLER PENALTIES OPEN AS P

;

HANDLER P READ AMOUNT\_PLAYERNO > (30.00, 44) LIMIT 100

Answer 16.1:

HANDLER MATCHES OPEN AS M1

;

HANDLER M1 READ FIRST

;

HANDLER M1 READ NEXT

;

HANDLER M1 CLOSE

Answer 16.2:

HANDLER MATCHES OPEN AS M2

;

HANDLER M2 READ `PRIMARY` FIRST

;

HANDLER M2 READ `PRIMARY` NEXT

;

HANDLER M2 CLOSE

Answer 16.3:

HANDLER MATCHES OPEN AS M3

;

HANDLER M3 READ `PRIMARY` LAST

WHERE PLAYERNO IN (6, 104, 112)

;

HANDLER M3 READ `PRIMARY` PREV

WHERE PLAYERNO IN (6, 104, 112)

;

HANDLER M3 CLOSE

Example 17.1:

INSERT INTO TEAMS (TEAMNO, PLAYERNO, DIVISION)

VALUES (3, 100, 'third')

;

INSERT INTO TEAMS

VALUES (3, 100, 'third')

;

INSERT INTO TEAMS (PLAYERNO, DIVISION, TEAMNO)

VALUES (100, 'third', 3)

;

INSERT INTO TEAMS

(TEAMNO, DIVISION)

VALUES (3, 'third')

Example 17.2:

INSERT INTO PLAYERS

(PLAYERNO, NAME, INITIALS, SEX,

JOINED, STREET, TOWN)

VALUES (611, 'Jones', 'GG', 'M', 1977, 'Green Way', 'Stratford')

;

INSERT INTO PLAYERS

(PLAYERNO, NAME, INITIALS, BIRTH\_DATE,

SEX, JOINED, STREET, HOUSENO, POSTCODE,

TOWN, PHONENO, LEAGUENO)

VALUES (611, 'Jones', 'GG', NULL, 'M', 1977,

'Green Way', NULL, NULL, 'Stratford', NULL, NULL)

Example 17.3:

INSERT INTO TEAMS (TEAMNO, PLAYERNO, DIVISION)

VALUES (6, 100, 'third'),

(7, 27, 'fourth'),

(8, 39, 'fourth'),

(9, 112, 'sixth')

Example 17.4:

CREATE TABLE TOTALS

(NUMBERPLAYERS INTEGER NOT NULL,

SUMPENALTIES DECIMAL(9,2) NOT NULL)

;

INSERT INTO TOTALS (NUMBERPLAYERS, SUMPENALTIES)

VALUES ((SELECT COUNT(\*) FROM PLAYERS),

(SELECT SUM(AMOUNT) FROM PENALTIES))

;

INSERT INTO TEAMS (TEAMNO, PLAYERNO, DIVISION)

VALUES (3, 100, 'third')

;

INSERT INTO TEAMS SET

TEAMNO = 3, PLAYERNO = 100, DIVISION = 'third'

Example 17.5:

INSERT IGNORE INTO TEAMS VALUES (1, 39, 'second')

Example 17.6:

INSERT INTO TEAMS VALUES (1, 39, 'second')

ON DUPLICATE KEY UPDATE PLAYERNO = 39, DIVISION='second'

Example 17.7:

CREATE TABLE RECR\_PLAYERS

(PLAYERNO SMALLINT NOT NULL,

NAME CHAR(15) NOT NULL,

TOWN CHAR(10) NOT NULL,

PHONENO CHAR(13),

PRIMARY KEY (PLAYERNO))

;

INSERT INTO RECR\_PLAYERS

(PLAYERNO, NAME, TOWN, PHONENO)

SELECT PLAYERNO, NAME, TOWN, PHONENO

FROM PLAYERS

WHERE LEAGUENO IS NULL

;

INSERT INTO RECR\_PLAYERS

SELECT PLAYERNO, NAME, TOWN, PHONENO

FROM PLAYERS

WHERE LEAGUENO IS NULL

;

INSERT INTO RECR\_PLAYERS

(TOWN, PHONENO, NAME, PLAYERNO)

SELECT TOWN, PHONENO, NAME, PLAYERNO

FROM PLAYERS

WHERE LEAGUENO IS NULL

Example 17.8:

INSERT INTO RECR\_PLAYERS

(PLAYERNO, NAME, TOWN, PHONENO)

SELECT PLAYERNO + 1000, NAME, TOWN, PHONENO

FROM RECR\_PLAYERS

Example 17.9:

INSERT INTO PENALTIES

SELECT PAYMENTNO + 100, PLAYERNO, PAYMENT\_DATE, AMOUNT

FROM PENALTIES

WHERE AMOUNT >

(SELECT AVG(AMOUNT)

FROM PENALTIES)

Example 17.10:

UPDATE PLAYERS

SET LEAGUENO = '2000'

WHERE PLAYERNO = 95

;

UPDATE PLAYERS AS P

SET P.LEAGUENO = '2000'

WHERE P.PLAYERNO = 95

Example 17.11:

UPDATE PENALTIES

SET AMOUNT = AMOUNT \* 1.05

Example 17.12:

UPDATE MATCHES

SET WON = 0

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PLAYERS

WHERE TOWN = 'Stratford')

Example 17.13:

UPDATE PLAYERS

SET STREET = 'Palmer Street',

HOUSENO = '83',

TOWN = 'Inglewood',

POSTCODE = '1234UU',

PHONENO = NULL

WHERE NAME = 'Parmenter'

;

UPDATE PLAYERS

SET STREET = TOWN,

TOWN = STREET

WHERE PLAYERNO = 44

;

UPDATE PLAYERS

SET STREET = TOWN

WHERE PLAYERNO = 44

;

UPDATE PLAYERS

SET TOWN = STREET

WHERE PLAYERNO = 44

Example 17.14:

CREATE TABLE PLAYERS\_DATA

(PLAYERNO INTEGER NOT NULL PRIMARY KEY,

NUMBER\_MAT INTEGER,

SUM\_PENALTIES DECIMAL(7,2))

;

INSERT INTO PLAYERS\_DATA (PLAYERNO)

SELECT PLAYERNO FROM PLAYERS

;

UPDATE PLAYERS\_DATA AS PD

SET NUMBER\_MAT = (SELECT COUNT(\*)

FROM MATCHES AS M

WHERE M.PLAYERNO = PD.PLAYERNO),

SUM\_PENALTIES = (SELECT SUM(AMOUNT)

FROM PENALTIES AS PEN

WHERE PEN.PLAYERNO = PD.PLAYERNO)

Example 17.15:

UPDATE PENALTIES

SET AMOUNT = AMOUNT – (SELECT AVG(AMOUNT)

FROM PENALTIES)

;

SET @AVERAGE\_AMOUNT = (SELECT AVG(AMOUNT) FROM PENALTIES)

;

UPDATE PENALTIES

SET AMOUNT = AMOUNT – @AVERAGE\_AMOUNT

Example 17.16:

UPDATE PENALTIES

SET AMOUNT = AMOUNT \* 1.05

ORDER BY AMOUNT DESC

Example 17.17:

UPDATE PENALTIES

SET PAYMENTNO = PAYMENTNO + 1

ORDER BY PAYMENTNO DESC

Example 17.18:

UPDATE PENALTIES

SET AMOUNT = AMOUNT \* 1.05

ORDER BY AMOUNT DESC, PLAYERNO ASC

LIMIT 4

Example 17.19:

UPDATE IGNORE MATCHES

SET MATCHNO = MATCHNO + 1,

WON = 2,

LOST = 3

WHERE MATCHNO = 4

Example 17.20:

UPDATE MATCHES AS M, TEAMS AS T

SET WON = 0

WHERE T.TEAMNO = M.TEAMNO

AND T.DIVISION = 'first'

;

SELECT ...

FROM MATCHES AS M, TEAMS AS T

WHERE T.TEAMNO = M.TEAMNO

AND T.DIVISION = 'first'

;

UPDATE MATCHES

SET WON = 0

WHERE TEAMNO IN

(SELECT TEAMNO

FROM TEAMS

WHERE DIVISION = 'first')

Example 17.21:

UPDATE MATCHES AS M, TEAMS AS T

SET M.WON = 0,

T.PLAYERNO = 112

WHERE T.TEAMNO = M.TEAMNO

AND T.DIVISION = 'first'

Example 17.22:

UPDATE PLAYERS AS P,

TEAMS AS T,

MATCHES AS M,

PENALTIES AS PEN,

COMMITTEE\_MEMBERS AS C

SET P.PLAYERNO = 1,

T.PLAYERNO = 1,

M.PLAYERNO = 1,

PEN.PLAYERNO = 1,

C.PLAYERNO = 1

WHERE P.PLAYERNO = T.PLAYERNO

AND T.PLAYERNO = M.PLAYERNO

AND M.PLAYERNO = PEN.PLAYERNO

AND PEN.PLAYERNO = C.PLAYERNO

AND C.PLAYERNO = 2

Example 17.23:

REPLACE INTO PLAYERS

(PLAYERNO, NAME, INITIALS, SEX,

JOINED, STREET, TOWN)

VALUES (611, 'Jones', 'GG', 'M', 1977, 'Green Way', 'Stratford')

Example 17.24:

REPLACE INTO TEAMS (TEAMNO, PLAYERNO, DIVISION)

VALUES (6, 100, 'third'),

(7, 27, 'fourth'),

(8, 39, 'fourth'),

(9, 112, 'sixth')

Example 17.25:

REPLACE INTO RECR\_PLAYERS

(PLAYERNO, NAME, TOWN, PHONENO)

SELECT PLAYERNO + 1000, NAME, TOWN, PHONENO

FROM RECR\_PLAYERS

Example 17.26:

DELETE

FROM PENALTIES

WHERE PLAYERNO = 44

;

DELETE

FROM PENALTIES AS PEN

WHERE PEN.PLAYERNO = 44

Example 17.27:

DELETE

FROM PLAYERS

WHERE JOINED >

(SELECT AVG(JOINED)

FROM PLAYERS

WHERE TOWN = 'Stratford')

Example 17.28:

DELETE

FROM PENALTIES

ORDER BY AMOUNT DESC, PLAYERNO ASC

LIMIT 4

Example 17.29:

DELETE IGNORE

FROM PLAYERS

Example 17.30:

DELETE MATCHES

FROM MATCHES, PLAYERS

WHERE MATCHES.PLAYERNO = PLAYERS.PLAYERNO

AND PLAYERS.TOWN = 'Inglewood'

;

SELECT ...

FROM MATCHES, PLAYERS

WHERE MATCHES.PLAYERNO = PLAYERS.PLAYERNO

AND PLAYERS.TOWN = 'Inglewood'

;

DELETE MATCHES

FROM MATCHES AS M, PLAYERS

WHERE M.PLAYERNO = PLAYERS.PLAYERNO

AND PLAYERS.TOWN = 'Inglewood'

Example 17.31:

DELETE TEAMS, MATCHES

FROM TEAMS, MATCHES

WHERE TEAMS.TEAMNO = MATCHES.TEAMNO

AND TEAMS.TEAMNO = 1

;

DELETE

FROM TEAMS, MATCHES

USING TEAMS, MATCHES

WHERE TEAMS.TEAMNO = MATCHES.TEAMNO

AND TEAMS.TEAMNO = 1

Example 17.32:

TRUNCATE TABLE COMMITTEE\_MEMBERS

Answer 17.1:

INSERT INTO PENALTIES

VALUES (15, 27, '1985-11-08', 75)

Answer 17.2:

INSERT INTO PENALTIES

SELECT PAYMENTNO + 1000, PLAYERNO, PAYMENT\_DATE, AMOUNT

FROM PENALTIES

WHERE AMOUNT >

(SELECT AVG(AMOUNT)

FROM PENALTIES)

UNION

SELECT PAYMENTNO + 2000, PLAYERNO, PAYMENT\_DATE, AMOUNT

FROM PENALTIES

WHERE PLAYERNO = 27

Answer 17.3:

UPDATE PLAYERS

SET SEX = 'W'

WHERE SEX = 'F'

Answer 17.4:

UPDATE PLAYERS

SET SEX = 'X'

WHERE SEX = 'F'

;

UPDATE PLAYERS

SET SEX = 'F'

WHERE SEX = 'M'

;

UPDATE PLAYERS

SET SEX = 'M'

WHERE SEX = 'X'

;

UPDATE PLAYERS

SET SEX = CASE SEX

WHEN 'F' THEN 'M'

ELSE 'F' END

Answer 17.5:

UPDATE PENALTIES

SET AMOUNT = AMOUNT \* 1.2

WHERE AMOUNT >

(SELECT AVG(AMOUNT)

FROM PENALTIES)

Answer 17.6:

UPDATE TEAMS AS T, PLAYERS AS P

SET DIVISION = 'third'

WHERE T.PLAYERNO = P.PLAYERNO

AND P.TOWN = 'Stratford'

Answer 17.7:

UPDATE PENALTIES, TEAMS

SET AMOUNT = 50,

DIVISION = 'fourth'

Answer 17.8:

DELETE

FROM PENALTIES

WHERE PLAYERNO = 44

AND YEAR(PAYMENT\_DATE) = 1980

Answer 17.9:

DELETE

FROM PENALTIES

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM MATCHES

WHERE TEAMNO IN

(SELECT TEAMNO

FROM TEAMS

WHERE DIVISION = 'second'))

Answer 17.10:

DELETE

FROM PLAYERS

WHERE TOWN =

(SELECT TOWN

FROM PLAYERS

WHERE PLAYERNO = 44)

AND PLAYERNO <> 44

Answer 17.11:

DELETE PEN, M

FROM PENALTIES AS PEN, MATCHES AS M

WHERE PEN.PLAYERNO = M.PLAYERNO

AND PEN.PLAYERNO = 27

Answer 17.12:

DELETE PEN, M

FROM PENALTIES AS PEN, MATCHES AS M

WHERE PEN.PLAYERNO = 27

AND M.PLAYERNO = 27

Example 18.1:

SELECT \*

FROM TEAMS

INTO OUTFILE 'C:/TEAMS.TXT'

Example 18.2:

SELECT \*

FROM TEAMS

INTO OUTFILE 'C:/TEAMS.TXT'

FIELDS TERMINATED BY ','

LINES TERMINATED BY '?'

Example 18.3:

SELECT \*

FROM TEAMS

INTO OUTFILE 'C:/TEAMS.TXT'

FIELDS TERMINATED BY ','

OPTIONALLY ENCLOSED BY '"'

LINES TERMINATED BY '?'

Example 18.4:

SELECT \*

FROM TEAMS

INTO OUTFILE 'C:/TEAMS.TXT'

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '?'

Example 18.5:

SELECT \*, NULL

FROM TEAMS

INTO OUTFILE 'C:/TEAMS.TXT'

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '?'

Example 18.6:

SELECT \*, NULL

FROM TEAMS

INTO OUTFILE 'C:/TEAMS.TXT'

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

ESCAPED BY '\*'

LINES TERMINATED BY '?'

Example 18.7:

SELECT \*

FROM TEAMS

INTO OUTFILE 'C:/TEAMS.TXT'

FIELDS TERMINATED BY ','

OPTIONALLY ENCLOSED BY '"'

LINES TERMINATED BY '\n'

Example 18.8:

SELECT \*

FROM TEAMS

INTO DUMPFILE 'C:/TEAMS.DUMP'

Example 18.9:

SELECT \*

FROM TEAMS

WHERE TEAMNO = 1

INTO @v1, @v2, @V3

;

SELECT @V1, @V2, @V3

Example 18.10:

LOAD DATA INFILE 'C:/TEAMS.TXT'

REPLACE

INTO TABLE TEAMS

FIELDS TERMINATED BY ','

LINES TERMINATED BY '?'

Example 18.11:

LOAD DATA INFILE 'C:/TEAMS.TXT'

REPLACE

INTO TABLE TEAMS

FIELDS TERMINATED BY ','

LINES TERMINATED BY '?'

IGNORE 1 LINES

Example 18.12:

LOAD DATA INFILE 'C:/TEAMS.TXT'

REPLACE

INTO TABLE TEAMS

FIELDS TERMINATED BY ','

LINES TERMINATED BY '?'

(PLAYERNO,TEAMNO,DIVISION)

;

SELECT \* FROM TEAMS

Example 18.13:

LOAD DATA INFILE 'C:/TEAMS.TXT'

REPLACE

INTO TABLE TEAMS

FIELDS TERMINATED BY ','

LINES TERMINATED BY '?'

SET DIVISION='xxx'

;

SELECT \* FROM TEAMS

Example 18.14:

LOAD DATA INFILE 'C:/TEAMS.TXT'

REPLACE

INTO TABLE TEAMS

FIELDS TERMINATED BY ','

LINES TERMINATED BY '?'

(TEAMNO,PLAYERNO,@DIV)

SET DIVISION=SUBSTRING(@DIV,1,1)

;

SELECT \* FROM TEAMS

Example 18.15:

LOAD DATA INFILE 'C:/TEAMS2.TXT'

REPLACE

INTO TABLE TEAMS

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\r'

STARTING BY '/\*/'

;

SELECT \* FROM TEAMS

Example 19.1:

CREATE TABLE XML\_MATCHES

(MATCHNO INTEGER NOT NULL PRIMARY KEY,

MATCH\_INFO TEXT)

Example 19.2:

INSERT INTO XML\_MATCHES VALUES (1,

'<match number=1>Match info of 1

<team>Team info of 1

<number>1</number>

<division>first</division>

</team>

<player>Player info of 6

<number>6</number>

<name>The name of 6

<lastname>Parmenter</lastname>

<initials>R</initials>

</name>

<address>The address of 6

<street>Haseltine Lane</street>

<houseno>80</houseno>

<postcode>1234KK</postcode>

<town>Stratford</town>

</address>

</player>

<sets>Info about sets of 1

<won>3</won>

<lost>1</lost>

</sets>

</match>')

;

INSERT INTO XML\_MATCHES VALUES (9,

'<match number=9>Match info of 9

<team>Team info of 2

<number>2</number>

<division>second</division>

</team>

<player>Player info of 27

<number>27</number>

<name>The name of 27

<lastname>Collins</lastname>

<initials>DD</initials>

</name>

<address>The address of 27

<street>Long Drive</street>

<houseno>804</houseno>

<postcode>8457DK</postcode>

<town>Eltham</town>

</address>

<phones>Phone numbers of 27

<number>1234567</number>

<number>3468346</number>

<number>6236984</number>

<number>6587437</number>

</phones>

</player>

<sets>Info about sets of 9

<won>3</won>

<lost>2</lost>

</sets>

</match>')

;

INSERT INTO XML\_MATCHES VALUES (12,

'<match number=12>Match info of 12

<team>Team info of 2

<number>2</number>

<division>second</division>

</team>

<player>Player info of 8

<number>8</number>

<name>The name of 8

<lastname>Newcastle</lastname>

<initials>B</initials>

</name>

<address>The first address van 8

<street>Station Road</street>

<houseno>4</houseno>

<postcode>6584RO</postcode>

<town>Inglewood</town>

</address>

<address>The second address of 8

<street>Trolley Lane</street>

<houseno>14</houseno>

<postcode>2728YG</postcode>

<town>Douglas</town>

</address>

</player>

<sets>Info about sets of 12

<won>1</won>

<lost>3</lost>

</sets>

</match>')

Example 19.3:

SELECT MATCHNO,

EXTRACTVALUE(MATCH\_INFO, '/match/team/division')

AS DIVISION

FROM XML\_MATCHES

Example 19.4:

SELECT MATCHNO,

EXTRACTVALUE(MATCH\_INFO,

'/match/player/name/lastname')

AS PLAYER

FROM XML\_MATCHES

WHERE EXTRACTVALUE(MATCH\_INFO,

'/match/sets/won') = 3

Example 19.5:

SELECT EXTRACTVALUE('

<team>

<number>2</number>

<division>second</division>

</team>'

,'/team') = '' AS TEAM

Example 19.6:

SELECT MATCHNO, EXTRACTVALUE(MATCH\_INFO,

'/match/player/phones/number')

AS PHONES

FROM XML\_MATCHES

WHERE MATCHNO = 9

Example 19.7:

SELECT MATCHNO, EXTRACTVALUE(MATCH\_INFO,

'/match/player')

AS PLAYERS

FROM XML\_MATCHES

;

SELECT REPLACE(EXTRACTVALUE(MATCH\_INFO,

'/match/player'), ' ', '#')

AS PLAYER\_INFO

FROM XML\_MATCHES

Example 19.8:

SELECT MATCHNO, EXTRACTVALUE(MATCH\_INFO,

'/match/\*/number')

AS NUMBERS

FROM XML\_MATCHES

Example 19.9:

SELECT MATCHNO, EXTRACTVALUE(MATCH\_INFO,

'/match//number')

AS NUMBERS

FROM XML\_MATCHES

;

SELECT MATCHNO,

EXTRACTVALUE(MATCH\_INFO, '//number')

AS NUMBERS

FROM XML\_MATCHES

Example 19.10:

SELECT EXTRACTVALUE(MATCH\_INFO, '/match//\*')

AS EVERYTHING

FROM XML\_MATCHES

WHERE MATCHNO = 1

Example 19.11:

SELECT MATCHNO, EXTRACTVALUE(MATCH\_INFO,

'//town|//won')

AS TOWN\_WON

FROM XML\_MATCHES

;

SELECT MATCHNO,

CONCAT(EXTRACTVALUE(MATCH\_INFO, '//town'),

' ',

EXTRACTVALUE(MATCH\_INFO, '//won'))

AS TOWN\_WON

FROM XML\_MATCHES

Example 19.12:

SELECT MATCHNO, EXTRACTVALUE(MATCH\_INFO,

'/match/@number')

AS XML\_MATCHNO

FROM XML\_MATCHES

Example 19.13:

SELECT MATCHNO, EXTRACTVALUE(MATCH\_INFO,

'/match/sets/won+10')

AS WON\_PLUS\_10

FROM XML\_MATCHES

Example 19.14:

SELECT MATCHNO, EXTRACTVALUE(MATCH\_INFO,

'/match/player/address[1]/town')

AS TOWN

FROM XML\_MATCHES

Example 19.15:

SELECT MATCHNO, EXTRACTVALUE(MATCH\_INFO,

'/match/player/\*[1]')

AS A\_VALUE

FROM XML\_MATCHES

Example 19.16:

SELECT MATCHNO, EXTRACTVALUE(MATCH\_INFO,

'/match/player/phones/number[last()]')

AS LAST

FROM XML\_MATCHES

Example 19.17:

SELECT MATCHNO, EXTRACTVALUE(MATCH\_INFO,

'/child::match/child::team/child::number ')

AS NUMBERS

FROM XML\_MATCHES

Example 19.18:

SELECT EXTRACTVALUE(MATCH\_INFO,

'/match/player/address/descendant::\* ')

AS ADDRESS\_INFO

FROM XML\_MATCHES

Example 19.19:

SELECT EXTRACTVALUE(MATCH\_INFO,

'/match/player/descendant::\* ')

AS PLAYER\_INFO

FROM XML\_MATCHES

Example 19.20:

SELECT EXTRACTVALUE(MATCH\_INFO,

'/match/player/descendant::\* ')

AS PLAYER\_INFO

FROM XML\_MATCHES

Example 19.21:

SELECT MATCHNO, EXTRACTVALUE(MATCH\_INFO,

'/match/player[number=8]')

AS PLAYER8

FROM XML\_MATCHES

Example 19.22:

SELECT MATCHNO, EXTRACTVALUE(MATCH\_INFO,

'/match/player')

AS PLAYER8

FROM XML\_MATCHES

WHERE EXTRACTVALUE(MATCH\_INFO,

'/match/player[number=8]') <> ''

Example 19.23:

SELECT MATCHNO, EXTRACTVALUE(MATCH\_INFO,

'/match/sets')

AS THREE\_AND\_ONE

FROM XML\_MATCHES

WHERE EXTRACTVALUE(MATCH\_INFO,

'/match/sets[won=3 and lost=1]') <>''

Example 19.24:

UPDATE XML\_MATCHES

SET MATCH\_INFO =

UPDATEXML(MATCH\_INFO,

'/match/sets/lost',

'<lost>2</lost>')

WHERE MATCHNO = 1

;

SELECT EXTRACTVALUE(MATCH\_INFO,

'/match/sets/lost') AS LOST

FROM XML\_MATCHES

WHERE MATCHNO = 1

Example 19.25:

UPDATE XML\_MATCHES

SET MATCH\_INFO =

UPDATEXML(MATCH\_INFO,

'/match/player/address',

'<address>The new address of 8

<street>Jolly Lane</street>

<houseno>30</houseno>

<postcode>5383GH</postcode>

<town>Douglas</town>

</address>')

WHERE MATCHNO = 1

;

SELECT EXTRACTVALUE(MATCH\_INFO,

'/match/player/address/\*') AS NEW\_ADDRESS

FROM XML\_MATCHES

WHERE MATCHNO = 1

Example 20.1:

CREATE TABLE PLAYERS

(PLAYERNO INTEGER NOT NULL PRIMARY KEY,

NAME CHAR(15) NOT NULL,

INITIALS CHAR(3) NOT NULL,

BIRTH\_DATE DATE NULL,

SEX CHAR(1) NOT NULL,

JOINED SMALLINT NOT NULL,

STREET VARCHAR(30) NOT NULL,

HOUSENO CHAR(4) NULL,

POSTCODE CHAR(6) NULL,

TOWN VARCHAR(30) NOT NULL,

PHONENO CHAR(13) NULL,

LEAGUENO CHAR(4) UNIQUE)

Example 20.2:

CREATE TABLE TEST.PENALTIES

(PAYMENTNO INTEGER NOT NULL PRIMARY KEY,

PLAYERNO INTEGER NOT NULL,

PAYMENT\_DATE DATE NOT NULL,

AMOUNT DECIMAL(7,2) NOT NULL)

Example 20.3:

CREATE TABLE WIDTH (C4 INTEGER(4))

;

INSERT INTO WIDTH VALUES (1)

Example 20.4:

CREATE TABLE MEASUREMENTS

(NR INTEGER, MEASUREMENT\_VALUE FLOAT(1))

;

INSERT INTO MEASUREMENTS VALUES

(1, 99.99),

(2, 99999.99),

(3, 99999999.99),

(4, 99999999999.99),

(5, 99999999999999.99),

(6, 0.999999),

(7, 0.9999999),

(8, 99999999.9999),

(9, (1.0/3))

;

SELECT \* FROM MEASUREMENTS

Example 20.5:

CREATE TABLE MEASUREMENTS

(NR INTEGER, MEASUREMENT\_VALUE FLOAT(10,3))

;

INSERT INTO MEASUREMENTS VALUES

(1, 99.99),

(2, 99999.99),

(3, 99999999.99),

(4, 99999999999.99),

(5, 99999999999999.99),

(6, 0.999999),

(7, 0.9999999),

(8, 99999999.9999),

(9, (1.0/3))

;

SELECT \* FROM MEASUREMENTS

Example 20.6:

CREATE TABLE PENALTIESDEF

(PAYMENTNO INTEGER UNSIGNED NOT NULL PRIMARY KEY,

PLAYERNO INTEGER UNSIGNED NOT NULL,

PAYMENT\_DATE DATE NOT NULL,

AMOUNT DECIMAL(7,2) NOT NULL)

Example 20.7:

CREATE TABLE WIDTH (C4 INTEGER(4) ZEROFILL)

;

INSERT INTO WIDTH VALUES (1)

;

INSERT INTO WIDTH VALUES (200)

Example 20.8:

CREATE TABLE PENALTIES

(PAYMENTNO INTEGER NOT NULL PRIMARY KEY,

PLAYERNO INTEGER NOT NULL,

PAYMENT\_DATE DATE NOT NULL,

AMOUNT DECIMAL(7,2) ZEROFILL NOT NULL)

;

SELECT AMOUNT FROM PENALTIES

Example 20.9:

CREATE TABLE MEASUREMENTS (NO INTEGER,

MEASUREMENT\_VALUE FLOAT(19,3) ZEROFILL)

;

INSERT INTO MEASUREMENTS VALUES

(1, 99.99),

(2, 99999.99),

(3, 99999999.99),

(4, 99999999999.99),

(5, 99999999999999.99),

(6, 0.999999),

(7, 0.9999999),

(8, 99999999.9999),

(9, (1.0/3))

;

SELECT \* FROM MEASUREMENTS

Example 20.10:

CREATE TABLE CITY\_NAMES

(SEQNO INTEGER UNSIGNED AUTO\_INCREMENT

NOT NULL PRIMARY KEY,

NAME VARCHAR(30) NOT NULL)

Example 20.11:

INSERT INTO CITY\_NAMES VALUES (NULL, 'London')

;

INSERT INTO CITY\_NAMES VALUES (NULL, 'New York')

;

INSERT INTO CITY\_NAMES (NAME) VALUES ('Paris')

;

SELECT \* FROM CITY\_NAMES

Example 20.12:

INSERT INTO CITY\_NAMES VALUES (8, 'Bonn')

;

INSERT INTO CITY\_NAMES VALUES (NULL, 'Amsterdam')

;

SELECT \* FROM CITY\_NAMES

Example 20.13:

DELETE FROM CITY\_NAMES

;

INSERT INTO CITY\_NAMES VALUES (NULL, 'Phoenix')

;

INSERT INTO CITY\_NAMES VALUES (NULL, 'Rome')

Example 20.14:

SET @@AUTO\_INCREMENT\_OFFSET = 10,

@@AUTO\_INCREMENT\_INCREMENT = 10

;

CREATE TABLE T10

(SEQNO INTEGER AUTO\_INCREMENT NOT NULL PRIMARY KEY)

;

INSERT INTO T10 VALUES (NULL),(NULL)

;

SELECT \* FROM T10

Example 20.15:

CREATE TEMPORARY TABLE SUMPENALTIES

(TOTAL DECIMAL(10,2))

;

INSERT INTO SUMPENALTIES

SELECT SUM(AMOUNT)

FROM PENALTIES

Example 20.16:

CREATE TABLE TESTTABLE (C1 INTEGER)

;

INSERT INTO TESTTABLE VALUES (1)

;

CREATE TEMPORARY TABLE TESTTABLE (C1 INTEGER, C2 INTEGER)

;

INSERT INTO TESTTABLE VALUES (2, 3)

;

SELECT \* FROM TESTTABLE

Example 20.17:

CREATE TABLE IF NOT EXISTS TEAMS

(TEAMNO INTEGER NOT NULL PRIMARY KEY,

PLAYERNO INTEGER NOT NULL,

DIVISION CHAR(6) NOT NULL)

Example 20.18:

CREATE TABLE TEAMS\_COPY1 LIKE TEAMS

Example 20.19:

CREATE TABLE TEAMS\_COPY2 AS

(SELECT \*

FROM TEAMS)

Example 20.20:

CREATE TABLE TEAMS\_COPY3 AS

(SELECT TEAMNO AS TNO, PLAYERNO AS PNO, DIVISION

FROM TEAMS)

;

SELECT \*

FROM TEAMS\_COPY3

Example 20.21:

CREATE TABLE TEAMS\_COPY4 AS

(SELECT TEAMNO, PLAYERNO

FROM TEAMS

WHERE PLAYERNO = 27)

Example 20.22:

CREATE TEMPORARY TABLE TEAMS AS

(SELECT \*

FROM TEAMS

Example 20.23:

CREATE TABLE TEAMS\_COPY5

(TEAMNO INTEGER NOT NULL PRIMARY KEY,

PLAYERNO INTEGER NULL,

DIVISION CHAR(10) NOT NULL) AS

(SELECT \*

FROM TEAMS)

;

CREATE TABLE TEAMS\_COPY5

(PLAYERNO INTEGER NULL,

DIVISION CHAR(10) NOT NULL) AS

(SELECT \*

FROM TEAMS)

Example 20.24:

CREATE TABLE TEAMS\_COPY6

(PLAYERNO INTEGER NULL,

COMMENT VARCHAR(100)) AS

(SELECT \*

FROM TEAMS)

;

SELECT \* FROM TEAMS\_COPY6

Example 20.25:

CREATE TABLE TEAMS\_COPY7

(TEAMNO INTEGER NOT NULL PRIMARY KEY)

REPLACE AS

(SELECT \* FROM TEAMS

UNION ALL

SELECT 2, 27, 'third'

ORDER BY 1, 3 DESC)

;

SELECT \* FROM TEAMS\_COPY7

Example 20.26:

CREATE TABLE PENALTIES

(PAYMENTNO INTEGER NOT NULL PRIMARY KEY,

PLAYERNO INTEGER NOT NULL,

PAYMENT\_DATE DATE NOT NULL DEFAULT '1990-01-01',

AMOUNT DECIMAL(7,2) NOT NULL DEFAULT 50.00)

;

INSERT INTO PENALTIES

(PAYMENTNO, PLAYERNO)

VALUES (15, 27)

;

INSERT INTO PENALTIES

(PAYMENTNO, PLAYERNO, PAYMENT\_DATE, AMOUNT)

VALUES (15, 27, DEFAULT, DEFAULT)

Example 20.27:

UPDATE PENALTIES

SET AMOUNT = DEFAULT

Example 20.28:

UPDATE PENALTIES

SET AMOUNT = YEAR(DEFAULT(PAYMENT\_DATE))\*10

Example 20.29:

CREATE TABLE PENALTIES

(PAYMENTNO INTEGER NOT NULL PRIMARY KEY

COMMENT 'Primary key of the table',

PLAYERNO INTEGER NOT NULL

COMMENT 'Player who has incurred the penalty',

PAYMENT\_DATE DATE NOT NULL

COMMENT 'Date on which the penalty has been paid',

AMOUNT DECIMAL(7,2) NOT NULL

COMMENT 'Amount of the penalty in dollars')

;

SELECT COLUMN\_NAME, COLUMN\_COMMENT

FROM INFORMATION\_SCHEMA.COLUMNS

WHERE TABLE\_NAME = 'PENALTIES'

Example 20.30:

SHOW ENGINES

Example 20.31:

CREATE TABLE SEXES

(SEX CHAR(1) NOT NULL PRIMARY KEY)

ENGINE = MYISAM

Example 20.32:

SELECT TABLE\_NAME, ENGINE

FROM INFORMATION\_SCHEMA.TABLES

WHERE TABLE\_NAME IN ('PLAYERS', 'PENALTIES', 'SEXES')

Example 20.33:

CREATE TEMPORARY TABLE SUMPENALTIES

(TOTAL DECIMAL(10,2))

ENGINE = MEMORY

Example 20.34:

CREATE TABLE PENALTIES\_1990

(PAYMENTNO INTEGER NOT NULL PRIMARY KEY)

ENGINE=MYISAM

;

INSERT INTO PENALTIES\_1990 VALUES (1),(2),(3)

;

CREATE TABLE PENALTIES\_1991

(PAYMENTNO INTEGER NOT NULL PRIMARY KEY)

ENGINE=MYISAM

;

INSERT INTO PENALTIES\_1991 VALUES (4),(5),(6)

;

CREATE TABLE PENALTIES\_1992

(PAYMENTNO INTEGER NOT NULL PRIMARY KEY)

ENGINE=MYISAM

;

INSERT INTO PENALTIES\_1992 VALUES (7),(8),(9);

;

CREATE TABLE PENALTIES\_ALL

(PAYMENTNO INTEGER NOT NULL PRIMARY KEY)

ENGINE = MERGE

UNION = (PENALTIES\_1990,PENALTIES\_1991,PENALTIES\_1992)

INSERT\_METHOD = NO

;

SELECT \* FROM PENALTIES\_ALL

Example 20.35:

CREATE VIEW PENALTIES\_ALL AS

SELECT \* FROM PENALTIES\_1990

UNION

SELECT \* FROM PENALTIES\_1991

UNION

SELECT \* FROM PENALTIES\_1992

Example 20.36:

CREATE TABLE CITY\_NAMES

(SEQNO INTEGER AUTO\_INCREMENT NOT NULL PRIMARY KEY,

NAME VARCHAR(30) NOT NULL)

AUTO\_INCREMENT = 10

;

INSERT INTO CITY\_NAMES VALUES (NULL, 'London')

;

INSERT INTO CITY\_NAMES VALUES (NULL, 'New York')

;

INSERT INTO CITY\_NAMES VALUES (NULL, 'Paris')

;

SELECT \* FROM CITY\_NAMES

Example 20.37:

CREATE TABLE PENALTIES

(PAYMENTNO INTEGER NOT NULL PRIMARY KEY

COMMENT 'Primary key of the table',

PLAYERNO INTEGER NOT NULL

COMMENT 'Player who has incurred the penalty',

PAYMENT\_DATE DATE NOT NULL

COMMENT 'Date on which the penalty has been paid',

AMOUNT DECIMAL(7,2) NOT NULL

COMMENT 'Sum of the penalty in Euro''s')

COMMENT = 'Penalties that have been paid by the tennis club'

;

SELECT TABLE\_NAME, TABLE\_COMMENT

FROM INFORMATION\_SCHEMA.TABLES

WHERE TABLE\_NAME = 'PENALTIES'

Example 20.38:

CREATE TABLE MATCHES

(MATCHNO INTEGER NOT NULL PRIMARY KEY,

TEAMNO INTEGER NOT NULL,

PLAYERNO INTEGER NOT NULL,

WON SMALLINT NOT NULL,

LOST SMALLINT NOT NULL)

AVG\_ROW\_LENGTH = 15

MAX\_ROWS = 2000000

MIN\_ROWS = 1000000

Example 20.39:

CREATE TABLE TEAMS\_CSV

(TEAMNO INTEGER NOT NULL,

PLAYERNO INTEGER NOT NULL,

DIVISION CHAR(6) NOT NULL)

ENGINE = CSV

;

INSERT INTO TEAMS\_CSV VALUES (1, 6, 'first')

;

INSERT INTO TEAMS\_CSV VALUES (2, 27, 'second')

Example 20.40:

SELECT \*

FROM MATCHES

INTO OUTFILE 'C:/MATCHES\_EXTERN.TXT'

FIELDS TERMINATED BY ',' ENCLOSED BY '"'

;

CREATE TABLE MATCHES\_CSV

(MATCHNO INTEGER NOT NULL,

TEAMNO INTEGER NOT NULL,

PLAYERNO INTEGER NOT NULL,

WON SMALLINT NOT NULL,

LOST SMALLINT NOT NULL)

ENGINE = CSV

;

FLUSH TABLE MATCHES\_CSV

;

SELECT \*

FROM MATCHES\_CSV

WHERE MATCHNO <= 4

Example 20.41:

SELECT COLUMN\_NAME, DATA\_TYPE, CHAR\_LENGTH, NULLABLE

FROM COLUMNS

WHERE TABLE\_NAME = 'PLAYERS'

AND TABLE\_CREATOR = 'TENNIS'

ORDER BY COLUMN\_NO

Example 20.42:

SELECT 'PLAYERS' AS TABLE\_NAME, COUNT(\*) AS NUMBER\_ROWS,

(SELECT COUNT(\*)

FROM COLUMNS

WHERE TABLE\_NAME = 'PLAYERS'

AND TABLE\_CREATOR = 'TENNIS') AS P

FROM PLAYERS

UNION

SELECT 'TEAMS', COUNT(\*),

(SELECT COUNT(\*)

FROM COLUMNS

WHERE TABLE\_NAME = 'TEAMS'

AND TABLE\_CREATOR = 'TENNIS') AS T

FROM TEAMS

UNION

SELECT 'PENALTIES', COUNT(\*),

(SELECT COUNT(\*)

FROM COLUMNS

WHERE TABLE\_NAME = 'PENALTIES'

AND TABLE\_CREATOR = 'TENNIS') AS PEN

FROM PENALTIES

UNION

SELECT 'MATCHES', COUNT(\*),

(SELECT COUNT(\*)

FROM COLUMNS

WHERE TABLE\_NAME = 'MATCHES'

AND TABLE\_CREATOR = 'TENNIS') AS M

FROM MATCHES

UNION

SELECT 'COMMITTEE\_MEMBERS', COUNT(\*),

(SELECT COUNT(\*)

FROM COLUMNS

WHERE TABLE\_NAME = 'COMMITTEE\_MEMBERS'

AND TABLE\_CREATOR = 'TENNIS') AS CM

FROM COMMITTEE\_MEMBERS

ORDER BY 1

Answer 20.6:

CREATE TABLE DEPARTMENT

( DEPNO CHAR(5) NOT NULL PRIMARY KEY,

BUDGET DECIMAL(8,2),

LOCATION VARCHAR(30))

Answer 20.7:

CREATE TABLE P\_COPY LIKE PLAYERS

Answer 20.8:

CREATE TABLE P2\_COPY AS (SELECT \* FROM PLAYERS)

Answer 20.9:

CREATE TABLE NUMBERS AS

(SELECT PLAYERNO

FROM PLAYERS

WHERE TOWN = 'Stratford')

Example 21.2:

CREATE TABLE DIPLOMAS

(STUDENT INTEGER NOT NULL,

COURSE INTEGER NOT NULL,

DDATE DATE NOT NULL,

SUCCESSFUL CHAR(1),

LOCATION VARCHAR(50),

PRIMARY KEY (STUDENT, COURSE, DDATE))

Example 21.3:

CREATE TABLE DIPLOMAS

(STUDENT INTEGER NOT NULL,

COURSE INTEGER NOT NULL,

DDATE DATE NOT NULL,

SUCCESSFUL CHAR(1),

LOCATION VARCHAR(50),

PRIMARY KEY INDEX\_PRIM (STUDENT, COURSE, DDATE))

Example 21.4:

CREATE TABLE TEAMS

(TEAMNO INTEGER NOT NULL,

PLAYERNO INTEGER NOT NULL UNIQUE,

DIVISION CHAR(6) NOT NULL,

PRIMARY KEY (TEAMNO))

;

CREATE TABLE TEAMS

(TEAMNO INTEGER NOT NULL,

PLAYERNO INTEGER NOT NULL,

DIVISION CHAR(6) NOT NULL,

PRIMARY KEY (TEAMNO),

UNIQUE (PLAYERNO))

Example 21.5:

CREATE TABLE PLAYERS

(PLAYERNO INTEGER NOT NULL,

NAME CHAR(15) NOT NULL,

INITIALS CHAR(3) NOT NULL,

BIRTH\_DATE DATE,

SEX CHAR(1) NOT NULL,

JOINED SMALLINT NOT NULL,

STREET VARCHAR(30) NOT NULL,

HOUSENO CHAR(4),

POSTCODE CHAR(6),

TOWN VARCHAR(30) NOT NULL,

PHONENO CHAR(13),

LEAGUENO CHAR(4) UNIQUE,

PRIMARY KEY (PLAYERNO))

Example 21.6:

SET @@STORAGE\_ENGINE = 'InnoDB'

;

CREATE TABLE TEAMS

(TEAMNO INTEGER NOT NULL,

PLAYERNO INTEGER NOT NULL,

DIVISION CHAR(6) NOT NULL,

PRIMARY KEY (TEAMNO),

FOREIGN KEY (PLAYERNO)

REFERENCES PLAYERS (PLAYERNO))

;

SELECT \*

FROM TEAMS

WHERE PLAYERNO NOT IN

(SELECT PLAYERNO

FROM PLAYERS)

Example 21.7:

CREATE TABLE TEAMS

(TEAMNO INTEGER NOT NULL,

PLAYERNO INTEGER NOT NULL,

DIVISION CHAR(6) NOT NULL,

PRIMARY KEY (TEAMNO),

FOREIGN KEY (PLAYERNO) REFERENCES PLAYERS (PLAYERNO))

Example 21.8:

CREATE TABLE MATCHES

(MATCHNO INTEGER NOT NULL,

TEAMNO INTEGER NOT NULL,

PLAYERNO INTEGER NOT NULL,

WON INTEGER NOT NULL,

LOST INTEGER NOT NULL,

PRIMARY KEY (MATCHNO),

FOREIGN KEY (TEAMNO) REFERENCES TEAMS (TEAMNO),

FOREIGN KEY (PLAYERNO) REFERENCES PLAYERS (PLAYERNO))

Example 21.9:

CREATE TABLE PENALTIES

(PAYMENTNO INTEGER NOT NULL,

PLAYERNO INTEGER NOT NULL,

PAYMENT\_DATE DATE NOT NULL,

AMOUNT DECIMAL(7,2) NOT NULL,

PRIMARY KEY (PAYMENTNO),

FOREIGN KEY (PLAYERNO) REFERENCES PLAYERS (PLAYERNO))

;

CREATE TABLE EMPLOYEES

(EMPLOYEE\_NO CHAR(10) NOT NULL,

MANAGER\_NO CHAR(10),

PRIMARY KEY (EMPLOYEE\_NO),

FOREIGN KEY (MANAGER\_NO)

REFERENCES EMPLOYEES (EMPLOYEE\_NO))

Example 21.10:

CREATE TABLE PENALTIES

(PAYMENTNO INTEGER NOT NULL,

PLAYERNO INTEGER NOT NULL,

PAYMENT\_DATE DATE NOT NULL,

AMOUNT DECIMAL(7,2) NOT NULL,

PRIMARY KEY (PAYMENTNO),

FOREIGN KEY (PLAYERNO) REFERENCES PLAYERS (PLAYERNO)

ON UPDATE RESTRICT

ON DELETE RESTRICT)

Example 21.11:

CREATE TABLE PENALTIES

(PAYMENTNO INTEGER NOT NULL,

PLAYERNO INTEGER NOT NULL,

PAYMENT\_DATE DATE NOT NULL,

AMOUNT DECIMAL(7,2) NOT NULL,

PRIMARY KEY (PAYMENTNO),

FOREIGN KEY (PLAYERNO) REFERENCES PLAYERS (PLAYERNO)

ON DELETE CASCADE)

;

DELETE

FROM PLAYERS

WHERE PLAYERNO = 127

;

DELETE

FROM PENALTIES

WHERE PLAYERNO = 127

Example 21.12:

CREATE TABLE PENALTIES

(PAYMENTNO INTEGER NOT NULL,

PLAYERNO INTEGER NOT NULL,

PAYMENT\_DATE DATE NOT NULL,

AMOUNT DECIMAL(7,2) NOT NULL,

PRIMARY KEY (PAYMENTNO),

FOREIGN KEY (PLAYERNO) REFERENCES PLAYERS (PLAYERNO)

ON DELETE SET NULL)

Example 21.13:

CREATE TABLE PLAYERS\_X

(PLAYERNO INTEGER NOT NULL,

SEX CHAR(1) NOT NULL

CHECK(SEX IN ('M', 'F')))

Example 21.14:

CREATE TABLE PLAYERS\_Y

(PLAYERNO INTEGER NOT NULL,

BIRTH\_DATE DATE NOT NULL

CHECK(BIRTH\_DATE > '1920-01-01'))

Example 21.15:

CREATE TABLE PLAYERS\_Z

(PLAYERNO SMALLINT NOT NULL,

BIRTH\_DATE DATE,

JOINED SMALLINT NOT NULL,

CHECK(YEAR(BIRTH\_DATE) < JOINED))

;

CREATE TABLE PLAYERS\_W

(PLAYERNO SMALLINT,

BIRTH\_DATE DATE NOT NULL,

JOINED SMALLINT NOT NULL,

CHECK(YEAR(BIRTH\_DATE) < JOINED),

CHECK(BIRTH\_DATE > '1920-01-01'),

CHECK(JOINED < 1880))

Example 21.16:

CREATE TABLE PLAYERS\_V

(PLAYERNO SMALLINT NOT NULL,

SEX CHAR(1) NOT NULL

CHECK(SEX IN

(SELECT SEX FROM PLAYERS)))

Example 21.17:

CREATE TABLE DIPLOMAS

(STUDENT INTEGER NOT NULL,

COURSE INTEGER NOT NULL,

DDATE DATE NOT NULL,

SUCCESSFUL CHAR(1),

LOCATION VARCHAR(50),

CONSTRAINT PRIMARY\_KEY\_DIPLOMAS

PRIMARY KEY (STUDENT, COURSE, DDATE))

Example 21.18:

CREATE TABLE PLAYERS

(PLAYERNO INTEGER NOT NULL,

NAME CHAR(15) NOT NULL,

INITIALS CHAR(3) NOT NULL,

BIRTH\_DATE DATE,

SEX CHAR(1) NOT NULL,

JOINED SMALLINT NOT NULL,

STREET VARCHAR(30) NOT NULL,

HOUSENO CHAR(4),

POSTCODE CHAR(6),

TOWN VARCHAR(30) NOT NULL,

PHONE CHAR(13),

LEAGUENO CHAR(4),

CONSTRAINT PRIMARY\_KEY\_PLAYERS

PRIMARY KEY(PLAYERNO),

CONSTRAINT JOINED

CHECK(JOINED > 1969),

CONSTRAINT POSTCODE\_SIX\_CHARACTERS\_LONG

CHECK(POSTCODE LIKE '\_\_\_\_\_\_'),

CONSTRAINT ALLOWED\_VALUES\_SEX

CHECK(SEX IN ('M', 'F')))

Answer 21.3:

CREATE TABLE MATCHES

(MATCHNO INTEGER NOT NULL,

TEAMNO INTEGER NOT NULL,

PLAYERNO INTEGER NOT NULL,

WON INTEGER NOT NULL,

LOST INTEGER NOT NULL,

PRIMARY KEY (MATCHNO))

;

CREATE TABLE MATCHES

(MATCHNO INTEGER NOT NULL PRIMARY KEY,

TEAMNO INTEGER NOT NULL,

PLAYERNO INTEGER NOT NULL,

WON INTEGER NOT NULL,

LOST INTEGER NOT NULL)

Example 22.1:

SHOW CHARACTER SET

;

SELECT CHARACTER\_SET\_NAME, DESCRIPTION,

DEFAULT\_COLLATE\_NAME, MAXLEN

FROM INFORMATION\_SCHEMA.CHARACTER\_SETS

Example 22.2:

SHOW COLLATION LIKE 'utf8%'

;

SELECT \*

FROM INFORMATION\_SCHEMA.COLLATIONS

WHERE COLLATION\_NAME LIKE 'utf8%'

Example 22.3:

CREATE TABLE TABUCS2

(C1 CHAR(10) CHARACTER SET ucs2

NOT NULL PRIMARY KEY,

C2 VARCHAR(10) CHARACTER SET ucs2)

Example 22.4:

CREATE TABLE TABDEFKARSET

(C1 CHAR(10) NOT NULL,

C2 VARCHAR(10))

;

SELECT COLUMN\_NAME, CHARACTER\_SET\_NAME

FROM INFORMATION\_SCHEMA.COLUMNS

WHERE TABLE\_NAME = 'TABDEFKARSET'

Example 22.5:

CREATE TABLE TABUTF8

(C1 CHAR(10) NOT NULL,

C2 VARCHAR(10))

DEFAULT CHARACTER SET utf8

;

SELECT COLUMN\_NAME, CHARACTER\_SET\_NAME

FROM INFORMATION\_SCHEMA.COLUMNS

WHERE TABLE\_NAME = 'TABUTF8'

Example 22.6:

SELECT TABLE\_NAME, COLUMN\_NAME, COLLATION\_NAME

FROM INFORMATION\_SCHEMA.COLUMNS

WHERE TABLE\_NAME IN ('TABUCS2', 'TABDEFKARSET')

Example 22.7:

CREATE TABLE TABCOLLATE

(C1 CHAR(10)

CHARACTER SET utf8

COLLATE utf8\_romanian\_ci NOT NULL,

C2 VARCHAR(10)

CHARACTER SET utf8

COLLATE utf8\_spanish\_ci)

;

SELECT COLUMN\_NAME, CHARACTER\_SET\_NAME, COLLATION\_NAME

FROM INFORMATION\_SCHEMA.COLUMNS

WHERE TABLE\_NAME = 'TABCOLLATE'

Example 22.8:

CREATE TABLE TABDEFCOL

(C1 CHAR(10) NOT NULL,

C2 VARCHAR(10))

CHARACTER SET utf8

COLLATE utf8\_romanian\_ci

;

SELECT COLUMN\_NAME, CHARACTER\_SET\_NAME, COLLATION\_NAME

FROM INFORMATION\_SCHEMA.COLUMNS

WHERE TABLE\_NAME = 'TABDEFCOL'

Example 22.9:

CREATE TABLE TWOCHARSETS

(C1 CHAR(10) CHARACTER SET 'latin1' NOT NULL,

C2 VARCHAR(10) CHARACTER SET 'hp8')

;

INSERT INTO TWOCHARSETS VALUES ('A', 'A')

;

SELECT \*

FROM TWOCHARSETS

WHERE C1 = C2

Example 22.10:

CREATE TABLE TWOCOLL

(C1 CHAR(10) COLLATE 'latin1\_general\_ci' NOT NULL,

C2 VARCHAR(10) COLLATE 'latin1\_danish\_ci')

;

INSERT INTO TWOCOLL VALUES ('A', 'A')

;

SELECT \*

FROM TWOCOLL

WHERE C1 = C2

;

SELECT \*

FROM TWOCOLL

WHERE C1 COLLATE latin1\_danish\_ci = C2

;

SELECT \*

FROM TWOCOLL

WHERE C1 COLLATE utf8\_general\_ci = C2

Example 22.11:

SELECT \_utf8'database'

Example 22.12:

SELECT COLLATION(\_utf8'database'),

COLLATION(\_utf8'database' COLLATE utf8\_bin),

COLLATION((SELECT MAX(NAME) FROM PLAYERS))

Example 22.13:

SELECT CHARSET(\_utf8'database'),

CHARSET((SELECT MAX(NAME) FROM PLAYERS))

Example 22.14:

SELECT \_latin1'Muller' AS NAME

UNION

SELECT CONCAT('M', \_latin1 x'FC', 'ller')

ORDER BY NAME COLLATE latin1\_swedish\_ci

Example 22.15:

CREATE TABLE LETTERS

(SEQNO INTEGER NOT NULL PRIMARY KEY,

LETTER CHAR(1) CHARACTER SET UTF8 NOT NULL)

;

INSERT INTO LETTERS VALUES (1, 'e'), (2, 'é'),(3, 'ë')

;

SELECT LETTER, COUNT(\*)

FROM (SELECT LETTER COLLATE latin2\_czech\_cs AS LETTER

FROM LETTERS) AS LATIN2\_CZECH\_LETTERS

GROUP BY LETTER

;

SELECT LETTER, COUNT(\*)

FROM (SELECT LETTER COLLATE latin2\_croatian\_ci AS LETTER

FROM LETTERS) AS LATIN2\_CROATIAN\_LETTERS

GROUP BY LETTER

Example 22.16:

SELECT LETTER

FROM LETTERS

ORDER BY LETTER

Example 22.17:

SELECT COERCIBILITY('Rick' COLLATE latin1\_general\_ci) AS C0,

COERCIBILITY(TEAMNO) AS C2,

COERCIBILITY(USER()) AS C3,

COERCIBILITY('Rick') AS C4,

COERCIBILITY(NULL) AS C5

FROM TEAMS

WHERE TEAMNO = 1

Example 22.18:

SELECT @@COLLATION\_DATABASE

Example 22.19:

SHOW VARIABLES LIKE 'CHARACTER\_SET%'

Answer 22.2:

SELECT CHARACTER\_SET\_NAME, COUNT(\*)

FROM INFORMATION\_SCHEMA.COLLATIONS

GROUP BY CHARACTER\_SET\_NAME

Answer 22.3:

SELECT CHARSET((SELECT MAX(TOWN) FROM PLAYERS)),

COLLATION((SELECT MAX(TOWN) FROM PLAYERS))

Answer 22.4:

SELECT TOWN

FROM PLAYERS

ORDER BY TOWN COLLATE latin1\_danish\_ci

Example 23.1:

CREATE TABLE PLAYERS\_SMALL

(PLAYERNO INTEGER NOT NULL PRIMARY KEY,

NAME CHAR(15) NOT NULL,

INITIALS CHAR(3) NOT NULL,

BIRTH\_DATE DATE,

SEX ENUM ('M','F'))

Example 23.2:

INSERT INTO PLAYERS\_SMALL

VALUES (24, 'Jones', 'P', '1985-04-22', 'M')

;

INSERT INTO PLAYERS\_SMALL

VALUES (25, 'Marx', 'L', '1981-07-01', 'F')

;

INSERT INTO PLAYERS\_SMALL

VALUES (111, 'Cruise', 'T', '1982-11-11', 'm')

;

INSERT INTO PLAYERS\_SMALL

VALUES (199, 'Schroder', 'L', '1970-02-12', 'X')

;

INSERT INTO PLAYERS\_SMALL

VALUES (201, 'Lie', 'T', '1972-02-12', NULL)

;

SELECT \* FROM PLAYERS\_SMALL

Example 23.3:

SELECT PLAYERNO, SEX, SEX \* 1

FROM PLAYERS\_SMALL

Example 23.4:

SELECT PLAYERNO, SEX

FROM PLAYERS\_SMALL

ORDER BY SEX

Example 23.5:

CREATE TABLE SEXES

(SEX CHAR(1) NOT NULL PRIMARY KEY)

;

INSERT INTO SEXES VALUES ('M'),('F')

;

CREATE TABLE PLAYERS\_SMALL2

(PLAYERNO INTEGER NOT NULL PRIMARY KEY,

NAME CHAR(15) NOT NULL,

INITIALS CHAR(3) NOT NULL,

BIRTH\_DATE DATE,

SEX CHAR(1),

FOREIGN KEY (SEX) REFERENCES SEXES (SEX))

Example 23.6:

CREATE TABLE TEAMS\_NEW

(TEAMNO INTEGER NOT NULL PRIMARY KEY,

PLAYERNO INTEGER NOT NULL,

DIVISION SET ('first','second','third','fourth'))

Example 23.7:

INSERT INTO TEAMS\_NEW VALUES (1, 27, 'first')

;

INSERT INTO TEAMS\_NEW VALUES (2, 27, 'first,third')

;

INSERT INTO TEAMS\_NEW VALUES (3, 27, 'first,third,sixth')

;

INSERT INTO TEAMS\_NEW VALUES (4, 27, 'first,fifth')

;

INSERT INTO TEAMS\_NEW VALUES (5, 27, NULL)

;

INSERT INTO TEAMS\_NEW VALUES (6, 27, 7)

;

INSERT INTO TEAMS\_NEW VALUES (7, 27, CONV(1001,2,10))

;

SELECT \* FROM TEAMS\_NEW

Example 23.8:

SELECT TEAMNO, DIVISION \* 1, BIN(DIVISION \* 1)

FROM TEAMS\_NEW

Example 23.9:

CREATE TABLE SERIES\_NUMBERS

(NUMBERS SET

('1','2','3','4','5','6','7','8','9','10',

'11','12','13','14','15','16','17','18','19','20',

'21','22','23','24','25','26','27','28','29','30',

'31','32','33','34','35','36','37','38','39','40'))

;

INSERT INTO SERIES\_NUMBERS VALUES ('1'),('20'),('40')

;

SELECT NUMBERS, BIN(NUMBERS \* 1)

FROM SERIES\_NUMBERS

Example 23.10:

INSERT INTO TEAMS\_NEW VALUES (8, 27, 'eighth')

;

INSERT INTO TEAMS\_NEW VALUES (9, 27, '')

;

SELECT TEAMNO, DIVISION, DIVISION \* 1, BIN(DIVISION \* 1)

FROM TEAMS\_NEW

WHERE TEAMNO IN (8, 9)

Example 23.11:

SELECT TEAMNO

FROM TEAMS\_NEW

WHERE DIVISION = 'first'

Example 23.12:

SELECT TEAMNO, DIVISION

FROM TEAMS\_NEW

WHERE DIVISION & POWER(2,3-1) = POWER(2,3-1)

Example 23.13:

SELECT TEAMNO, DIVISION

FROM TEAMS\_NEW

WHERE DIVISION & 9 = 9

Example 23.14:

SELECT TEAMNO,

LENGTH(REPLACE(CONV((DIVISION \* 1),10,2),'0',''))

AS NUMBER

FROM TEAMS\_NEW

Example 23.15:

SELECT TEAMNO,

CASE WHEN (DIVISION & POWER(2,1-1) = POWER(2,1-1)) = 1

THEN 'YES' ELSE 'NO' END AS FIRST,

CASE WHEN (DIVISION & POWER(2,2-1) = POWER(2,2-1)) = 1

THEN 'YES' ELSE 'NO' END AS SECOND,

CASE WHEN (DIVISION & POWER(2,3-1) = POWER(2,3-1)) = 1

THEN 'YES' ELSE 'NO' END AS THIRD,

CASE WHEN (DIVISION & POWER(2,4-1) = POWER(2,4-1)) = 1

THEN 'YES' ELSE 'NO' END AS FOURTH

FROM TEAMS\_NEW

Example 23.16:

SELECT DIVISION, COUNT(\*)

FROM TEAMS\_NEW

WHERE DIVISION > 0

OR DIVISION IS NULL

GROUP BY DIVISION

Example 23.17:

UPDATE TEAMS\_NEW

SET DIVISION = DIVISION | POWER(2,3-1)

WHERE TEAMNO = 1

Example 23.18:

UPDATE TEAMS\_NEW

SET DIVISION = DIVISION & CONV(1011,2,10)

Example 23.19:

UPDATE TEAMS\_NEW

SET DIVISION = 0

Answer 23.1:

CREATE TABLE MATCHES

(MATCHNO INTEGER NOT NULL PRIMARY KEY,

TEAMNO INTEGER NOT NULL,

PLAYERNO INTEGER NOT NULL,

WON ENUM('0','1','2','3') NOT NULL,

LOST ENUM('0','1','2','3') NOT NULL)

Example 24.1:

DROP TABLE PLAYERS

Example 24.2:

DROP TABLE DB8.TAB1

Example 24.3:

DROP TABLES COMMITTEE\_MEMBERS, MATCHES, TEAMS,

PENALTIES, PLAYERS

Example 24.4:

RENAME TABLE PLAYERS TO TENNIS\_PLAYERS

Example 24.5:

RENAME TABLES PLAYERS TO TENNIS\_PLAYERS,

COMMITTEE\_MEMBERS TO MEMBERS

Example 24.6:

ALTER TABLE PLAYERS RENAME TO TENNIS\_PLAYERS

Example 24.7:

ALTER TABLE CITY\_NAMES

AUTO\_INCREMENT = 10000

COMMENT = 'New comment'

Example 24.8:

ALTER TABLE PLAYERS

CONVERT TO CHARACTER SET utf8 COLLATE utf8\_general\_ci

Example 24.9:

ALTER TABLE PLAYERS ORDER BY LEAGUENO DESC

Example 24.10:

ALTER TABLE TEAMS

ADD TYPE CHAR(1)

Example 24.11:

ALTER TABLE TEAMS

ADD TYPE CHAR(1) AFTER TEAMNO

Example 24.12:

ALTER TABLE TEAMS

ADD (CATEGORY VARCHAR(20) NOT NULL,

IMAGE INTEGER DEFAULT 10)

Example 24.13:

ALTER TABLE TEAMS

DROP TYPE

Example 24.14:

ALTER TABLE PLAYERS

CHANGE BIRTH\_DATE DATE\_OF\_BIRTH DATE

Example 24.15:

ALTER TABLE PLAYERS

CHANGE TOWN TOWN VARCHAR(40) NOT NULL

Example 24.16:

ALTER TABLE PLAYERS

CHANGE TOWN TOWN VARCHAR(5) NOT NULL

Example 24.17:

ALTER TABLE PLAYERS

CHANGE PLAYERNO PLAYERNO SMALLINT

Example 24.18:

ALTER TABLE PLAYERS

CHANGE TOWN TOWN VARCHAR(5) NOT NULL AFTER PLAYERNO

Example 24.19:

ALTER TABLE PLAYERS

MODIFY TOWN VARCHAR(5) NOT NULL AFTER PLAYERNO

Example 24.20:

ALTER TABLE COMMITTEE\_MEMBERS

ALTER POSITION SET DEFAULT 'Member'

;

ALTER TABLE COMMITTEE\_MEMBERS

MODIFY POSITION CHAR(20) DEFAULT 'Member'

Example 24.21:

ALTER TABLE COMMITTEE\_MEMBERS

ALTER POSITION DROP DEFAULT

Example 24.22:

CREATE TABLE T1

(A INTEGER NOT NULL PRIMARY KEY,

B INTEGER NOT NULL)

;

CREATE TABLE T2

(A INTEGER NOT NULL PRIMARY KEY,

B INTEGER NOT NULL,

CONSTRAINT C1 CHECK (B > 0),

CONSTRAINT FK1 FOREIGN KEY (A) REFERENCES T1 (A))

;

ALTER TABLE T1

ADD CONSTRAINT FK2 FOREIGN KEY (A) REFERENCES T2 (A)

Example 24.23:

ALTER TABLE PLAYERS DROP PRIMARY KEY

Example 24.24:

ALTER TABLE T1 DROP CONSTRAINT FK2

Example 24.25:

ALTER TABLE T2 DROP CONSTRAINT C1

Answer 24.1:

ALTER TABLE TEAMS

ENGINE = MYISAM

Answer 24.2:

ALTER TABLE COMMITTEE\_MEMBERS

ORDER BY PLAYERNO ASC, POSITION DESC

Answer 24.3:

ALTER TABLE COMMITTEE\_MEMBERS

CHANGE POSITION COMMITTEE\_POSITION CHAR(20)

Answer 24.4:

ALTER TABLE COMMITTEE\_MEMBERS

MODIFY COMMITTEE\_POSITION CHAR(30)

Answer 24.5:

ALTER TABLE PLAYERS

ALTER TOWN SET DEFAULT 'Stratford'

Example 25.4:

SELECT \*

FROM PLAYERS

WHERE PLAYERNO = 44

Example 25.5:

SELECT PLAYERNO, TOWN

FROM PLAYERS

WHERE PLAYERNO < 10

AND TOWN = 'Stratford'

ORDER BY PLAYERNO

Example 25.6:

SELECT NAME, INITIALS

FROM PLAYERS

WHERE TOWN =

(SELECT TOWN

FROM PLAYERS

WHERE PLAYERNO = 44)

Example 25.7:

CREATE INDEX PLAY\_PC

ON PLAYERS (POSTCODE ASC)

;

CREATE INDEX PLAY\_PC USING BTREE

ON PLAYERS (POSTCODE ASC)

Example 25.8:

CREATE INDEX PLAY\_TOWN USING HASH

ON PLAYERS (TOWN)

Example 25.9:

CREATE INDEX MAT\_WL

ON MATCHES (WON, LOST)

Example 25.10:

CREATE UNIQUE INDEX NAMEINIT

ON PLAYERS (NAME, INITIALS)

Example 25.11:

ALTER TABLE TEAMS

ADD INDEX TEAMS\_DIVISION USING BTREE (DIVISION)

Example 25.12:

ALTER TABLE PLAYERS

ADD UNIQUE INDEX TEAMS\_DIVISION

USING HASH (TOWN, STREET, BIRTH\_DATE)

Example 25.13:

CREATE TABLE MATCHES

(MATCHNO INTEGER NOT NULL PRIMARY KEY,

TEAMNO INTEGER NOT NULL,

PLAYERNO INTEGER NOT NULL,

WON SMALLINT NOT NULL,

LOST SMALLINT NOT NULL,

INDEX MAT\_WL (WON, LOST))

Example 25.14:

CREATE TABLE PLAYERS

(PLAYERNO INTEGER NOT NULL PRIMARY KEY,

NAME CHAR(15) NOT NULL,

INITIALS CHAR(3) NOT NULL,

BIRTH\_DATE DATE,

SEX CHAR(1) NOT NULL,

JOINED SMALLINT NOT NULL,

STREET VARCHAR(30) NOT NULL,

HOUSENO CHAR(4),

POSTCODE CHAR(6),

TOWN VARCHAR(30) NOT NULL,

PHONENO CHAR(13),

LEAGUENO CHAR(4),

UNIQUE INDEX NAMEINIT USING HASH (NAME, INITIALS))

Example 25.15:

DROP INDEX PLAY\_PC ON PLAYERS

;

DROP INDEX MATD\_WL ON MATCHES

;

DROP INDEX NAMEINIT ON PLAYERS

Example 25.16:

CREATE TABLE T1

(COL1 INTEGER NOT NULL,

COL2 DATE NOT NULL UNIQUE,

COL3 INTEGER NOT NULL,

COL4 INTEGER NOT NULL,

PRIMARY KEY (COL1, COL4),

UNIQUE (COL3, COL4),

UNIQUE (COL3, COL1))

;

CREATE UNIQUE INDEX "PRIMARY" USING BTREE

ON T1 (COL1, COL4)

;

CREATE UNIQUE INDEX COL2 USING BTREE

ON T1 (COL2)

;

CREATE UNIQUE INDEX COL3 USING BTREE

ON T1 (COL3, COL4)

;

CREATE UNIQUE INDEX COL3\_2 USING BTREE

ON T1 (COL3, COL1)

Example 25.17:

CREATE TABLE PLAYERS\_XXL

(PLAYERNO INTEGER NOT NULL PRIMARY KEY,

NAME CHAR(15) NOT NULL,

INITIALS CHAR(3) NOT NULL,

BIRTH\_DATE DATE,

SEX CHAR(1) NOT NULL,

JOINED SMALLINT NOT NULL,

STREET VARCHAR(30) NOT NULL,

HOUSENO CHAR(4),

POSTCODE CHAR(6),

TOWN VARCHAR(30) NOT NULL,

PHONENO CHAR(13),

LEAGUENO CHAR(8))

Example 25.18:

CREATE PROCEDURE FILL\_PLAYERS\_XXL

(IN NUMBER\_PLAYERS INTEGER)

BEGIN

DECLARE COUNTER INTEGER;

TRUNCATE TABLE PLAYERS\_XXL;

COMMIT WORK;

SET COUNTER = 1;

WHILE COUNTER <= NUMBER\_PLAYERS DO

INSERT INTO PLAYERS\_XXL VALUES(

COUNTER,

CONCAT('name',CAST(COUNTER AS CHAR(10))),

CASE MOD(COUNTER,2) WHEN 0 THEN 'vl1' ELSE 'vl2' END,

DATE('1960-01-01') + INTERVAL (MOD(COUNTER,300)) MONTH,

CASE MOD(COUNTER,20) WHEN 0 THEN 'F' ELSE 'M' END,

1980 + MOD(COUNTER,20),

CONCAT('street',CAST(COUNTER /10 AS UNSIGNED INTEGER)),

CAST(CAST(COUNTER /10 AS UNSIGNED INTEGER)+1 AS CHAR(4)),

CONCAT('p',MOD(COUNTER,50)),

CONCAT('town',MOD(COUNTER,10)),

'070-6868689',

CASE MOD(COUNTER,3) WHEN 0

THEN NULL ELSE cast(COUNTER AS CHAR(8)) END);

IF MOD(COUNTER,1000) = 0 THEN

COMMIT WORK;

END IF;

SET COUNTER = COUNTER + 1;

END WHILE;

COMMIT WORK;

END

Example 25.19:

CALL FILL\_PLAYERS\_XXL(100000)

Example 25.20:

CREATE INDEX PLAYERS\_XXL\_INITIALS

ON PLAYERS\_XXL(INITIALS)

;

CREATE INDEX PLAYERS\_XXL\_POSTCODE

ON PLAYERS\_XXL(POSTCODE)

;

CREATE INDEX PLAYERS\_XXL\_STREET

ON PLAYERS\_XXL(STREET)

Example 25.21:

SELECT TABLE\_CREATOR, TABLE\_NAME, COUNT(\*)

FROM INDEXES

GROUP BY TABLE\_CREATOR, TABLE\_NAME

HAVING COUNT(\*) > 1

Example 25.22:

SELECT TABLE\_CREATOR, TABLE\_NAME

FROM TABLES AS TAB

WHERE NOT EXISTS

(SELECT \*

FROM INDEXES AS IDX

WHERE TAB.TABLE\_CREATOR = IDX.TABLE\_CREATOR

AND TAB.TABLE\_NAME = TAB.TABLE\_NAME

AND IDX.UNIQUE\_ID = 'YES')

Example 25.23:

SHOW INDEX FROM PLAYERS

Example 26.1:

CREATE VIEW TOWNS AS

SELECT DISTINCT TOWN

FROM PLAYERS

;

SELECT \*

FROM TOWNS

Example 26.2:

CREATE VIEW CPLAYERS AS

SELECT PLAYERNO, LEAGUENO

FROM PLAYERS

WHERE LEAGUENO IS NOT NULL

;

SELECT \*

FROM CPLAYERS

Example 26.3:

SELECT \*

FROM CPLAYERS

WHERE PLAYERNO BETWEEN 6 AND 44

;

SELECT PLAYERNO, LEAGUENO

FROM PLAYERS

WHERE LEAGUENO IS NOT NULL

AND PLAYERNO BETWEEN 6 AND 44

Example 26.4:

DELETE

FROM CPLAYERS

WHERE LEAGUENO = '7060'

Example 26.5:

CREATE VIEW SEVERAL AS

SELECT \*

FROM CPLAYERS

WHERE PLAYERNO BETWEEN 6 AND 27

;

SELECT \*

FROM SEVERAL

Example 26.6:

CREATE VIEW DIGITS AS

SELECT 0 DIGIT UNION SELECT 1 UNION

SELECT 2 UNION SELECT 3 UNION

SELECT 4 UNION SELECT 5 UNION

SELECT 6 UNION SELECT 7 UNION

SELECT 8 UNION SELECT 9

;

SELECT \* FROM DIGITS

Example 26.7:

CREATE VIEW STRATFORDERS (PLAYERNO, NAME, INIT, BORN) AS

SELECT PLAYERNO, NAME, INITIALS, BIRTH\_DATE

FROM PLAYERS

WHERE TOWN = 'Stratford'

;

SELECT \*

FROM STRATFORDERS

WHERE PLAYERNO > 90

Example 26.8:

CREATE VIEW RESIDENTS AS

SELECT TOWN, COUNT(\*)

FROM PLAYERS

GROUP BY TOWN

;

SELECT TOWN, "COUNT(\*)"

FROM RESIDENTS

Example 26.9:

CREATE VIEW VETERANS AS

SELECT \*

FROM PLAYERS

WHERE BIRTH\_DATE < '1960-01-01'

;

UPDATE VETERANS

SET BIRTH\_DATE = '1970-09-01'

WHERE PLAYERNO = 2

;

CREATE VIEW VETERANS AS

SELECT \*

FROM PLAYERS

WHERE BIRTH\_DATE < '1960-01-01'

WITH CHECK OPTION

Example 26.10:

CREATE VIEW INGLEWOOD\_VETERANS AS

SELECT \*

FROM VETERANS

WHERE TOWN = 'Inglewood'

WITH CASCADED CHECK OPTION

Example 26.11:

CREATE DEFINER = 'JACO'@'%' VIEW JACO\_VIEW AS

SELECT \*

FROM PLAYERS

WHERE PLAYERNO > 100

Example 26.12:

CREATE SQL SECURITY INVOKER

ALGORITHM = MERGE

VIEW SIMPLE\_VIEW AS

SELECT PLAYERNO

FROM PLAYERS

WHERE PLAYERNO > 100

Example 26.13:

DROP VIEW CPLAYERS

Example 26.14:

SELECT TABLE\_NAME

FROM TABLES

WHERE TABLE\_NAME = 'STOCK'

AND TABLE\_CREATOR = 'TENNIS'

UNION

SELECT VIEW\_NAME

FROM VIEWS

WHERE VIEW\_NAME = 'STOCK'

AND VIEW\_CREATOR = 'TENNIS'

Example 26.15:

CREATE VIEW COST\_RAISERS AS

SELECT \*

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PENALTIES)

;

SELECT PLAYERNO

FROM COST\_RAISERS

WHERE TOWN = 'Stratford'

;

SELECT PLAYERNO

FROM (SELECT \*

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PENALTIES)) AS VIEWFORMULA

WHERE TOWN = 'Stratford'

Example 26.16:

DELETE

FROM STRATFORDERS

WHERE BORN > '1965-12-31'

;

DELETE

FROM PLAYERS

WHERE BIRTH\_DATE > '1965-12-31'

AND TOWN = 'Stratford'

;

SELECT \*

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PENALTIES)

Example 26.17:

CREATE VIEW EXPENSIVE\_PLAYERS AS

ALORITHM = TEMPTABLE

SELECT \*

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PENALTIES)

Example 26.18:

SELECT \*

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PENALTIES)

AND TOWN = 'Stratford'

;

SELECT TOWN, COUNT(\*)

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PENALTIES)

GROUP BY TOWN

;

CREATE VIEW PPLAYERS AS

SELECT \*

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PENALTIES)

;

SELECT \*

FROM PPLAYERS

WHERE TOWN = 'Stratford'

;

SELECT TOWN, COUNT(\*)

FROM PPLAYERS

GROUP BY TOWN

Example 26.20:

SELECT DISTINCT NAME, INITIALS, DIVISION

FROM PLAYERS AS P, MATCHES AS M, TEAMS AS T

WHERE P.PLAYERNO = M.PLAYERNO

AND M.TEAMNO = T.TEAMNO

;

CREATE VIEW TEAMS (TEAMNO, PLAYERNO, DIVISION) AS

SELECT DISTINCT TEAMNO, CAPTAIN, DIVISION

FROM RESULT

;

CREATE VIEW MATCHES AS

SELECT MATCHNO, TEAMNO, PLAYERNO,

WON, LOST

FROM RESULT

Example 26.21:

CREATE VIEW PLAYERSS AS

SELECT \*

FROM PLAYERS

WHERE SEX IN ('M', 'F')

WITH CHECK OPTION

Answer 26.1:

CREATE VIEW NUMBERPLS (TEAMNO, NUMBER) AS

SELECT TEAMNO, COUNT(\*)

FROM MATCHES

GROUP BY TEAMNO

Answer 26.2:

CREATE VIEW WINNERS AS

SELECT PLAYERNO, NAME

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM MATCHES

WHERE WON > LOST)

Answer 26.3:

CREATE VIEW TOTALS (PLAYERNO, SUM\_PENALTIES) AS

SELECT PLAYERNO, SUM(AMOUNT)

FROM PENALTIES

GROUP BY PLAYERNO

Answer 26.5.1:

SELECT YEAR(BORN) – 1900 AS DIFFERENCE, COUNT(\*)

FROM (SELECT PLAYERNO, NAME,

INITIALS, BIRTH\_DATE AS BORN

FROM PLAYERS

WHERE TOWN = 'Stratford') AS STRATFORDERS

GROUP BY DIFFERENCE

Answer 26.5.2:

SELECT EXPENSIVE.PLAYERNO

FROM (SELECT \*

FROM PLAYERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PENALTIES)) AS EXPENSIVE,

(SELECT PLAYERNO, NAME,

INITIALS, BIRTH\_DATE AS BORN

FROM PLAYERS

WHERE TOWN = 'Stratford') AS STRATFORDERS

WHERE EXPENSIVE.PLAYERNO = STRATFORDERS.PLAYERNO

Answer 26.5.3:

UPDATE PLAYERS

SET BIRTH\_DATE = '1950-04-04'

WHERE PLAYERNO = 7

Example 27.1:

SELECT SCHEMA\_NAME

FROM INFORMATION\_SCHEMA.SCHEMATA

Example 27.2:

SELECT TABLE\_NAME

FROM INFORMATION\_SCHEMA.TABLES

WHERE TABLE\_SCHEMA = 'TENNIS'

ORDER BY TABLE\_NAME

Example 27.3:

CREATE DATABASE TENNIS2

DEFAULT CHARACTER SET utf8

DEFAULT COLLATE utf8\_general\_ci

Example 27.4:

SELECT SCHEMA\_NAME, DEFAULT\_CHARACTER\_SET\_NAME,

DEFAULT\_COLLATION\_NAME

FROM INFORMATION\_SCHEMA.SCHEMATA

Example 27.5:

ALTER DATABASE TENNIS2

DEFAULT CHARACTER SET sjis

DEFAULT COLLATE sjis\_japanese\_ci

Example 27.6:

ALTER DATABASE TENNIS CHARACTER SET hp8

;

CREATE TABLE CHARSETHP8

(C1 CHAR(10) NOT NULL,

C2 VARCHAR(10))

;

SELECT COLUMN\_NAME, CHARACTER\_SET\_NAME, COLLATION\_NAME

FROM INFORMATION\_SCHEMA.COLUMNS

WHERE TABLE\_NAME = 'CHARSETHP8'

Example 27.7:

ALTER DATABASE TENNIS COLLATE hp8\_bin

Example 27.8:

DROP DATABASE TENNIS2

Example 28.1:

CREATE USER

'CHRIS'@'localhost' IDENTIFIED BY 'CHRISSEC',

'PAUL'@'localhost' IDENTIFIED BY 'LUAP'

Example 28.2:

CREATE USER

'CHRIS1'@'sql.r20.com' IDENTIFIED BY 'CHRISSEC1',

'CHRIS2'@'%' IDENTIFIED BY 'CHRISSEC2',

'CHRIS3'@'%.r20.com' IDENTIFIED BY 'CHRISSEC3'

;

SELECT \*

FROM USERS

WHERE USER\_NAME LIKE '''CHRIS%'

ORDER BY 1

Example 28.3:

DROP USER JIM

Example 28.4:

RENAME USER

'CHRIS1'@'sql.r20.com' TO 'COMBO1'@'sql.r20.com',

'CHRIS2'@'%' TO 'COMBO2'@'sql.r20.com'

;

SELECT \*

FROM USERS

WHERE USER\_NAME LIKE '''COMBO%'

ORDER BY 1

Example 28.5:

SET PASSWORD FOR 'JOHN'= PASSWORD('JOHN1')

Example 28.6:

SET PASSWORD FOR ROB = PASSWORD('ROBSEC')

Example 28.7:

GRANT SELECT

ON PLAYERS

TO JAMIE

Example 28.8:

GRANT SELECT

ON PLAYERS

TO 'JAMIE'@'localhost' IDENTIFIED BY 'JAMIEPASS'

Example 28.9:

GRANT INSERT, UPDATE

ON TEAMS

TO JAMIE, PIET

Example 28.10:

GRANT UPDATE (PLAYERNO, DIVISION)

ON TEAMS

TO PETE

Example 28.11:

GRANT SELECT

ON TENNIS.\*

TO PETE

Example 28.12:

GRANT CREATE, ALTER, DROP, CREATE VIEW

ON TENNIS.\*

TO JIM

Example 28.13:

GRANT SELECT

ON INFORMATION\_SCHEMA.\*

TO PETE

Example 28.14:

GRANT SELECT, INSERT

ON \*

TO ALYSSA

Example 28.15:

GRANT CREATE, ALTER, DROP

ON \*.\*

TO MAX

Example 28.16:

GRANT CREATE USER

ON \*.\*

TO ALYSSA

;

GRANT ALL PRIVILEGES

ON \*.\*

TO ROOT

Example 28.17:

GRANT REFERENCES

ON TEAMS

TO JOHN

WITH GRANT OPTION

;

GRANT REFERENCES

ON TEAMS

TO PETE

Example 28.18:

GRANT INSERT

ON COMMITTEE\_MEMBERS

TO MARC

;

GRANT SELECT

ON COMMITTEE\_MEMBERS

TO MARC

WITH GRANT OPTION

Example 28.19:

GRANT SELECT

ON \*.\*

TO SAM

WITH GRANT OPTION

Example 28.20:

GRANT SELECT

ON \*

TO JIM

WITH MAX\_QUERIES\_PER\_HOUR 1

Example 28.21:

SELECT GRANTEE

FROM USER\_AUTHS

WHERE PRIVILEGE = 'SELECT'

UNION

SELECT GRANTEE

FROM DATABASE\_AUTHS

WHERE DATABASE\_NAME = 'TENNIS'

AND PRIVILEGE = 'SELECT'

UNION

SELECT GRANTEE

FROM TABLE\_AUTHS

WHERE TABLE\_CREATOR = 'TENNIS'

AND PRIVILEGE = 'SELECT'

AND TABLE\_NAME = 'PLAYERS'

Example 28.22:

REVOKE SELECT

ON PLAYERS

FROM JIM

Example 28.23:

REVOKE REFERENCES

ON TEAMS

FROM JIM

Example 28.24:

REVOKE INSERT, SELECT

ON COMMITTEE\_MEMBERS

FROM MARC

;

REVOKE GRANT OPTION

ON COMMITTEE\_MEMBERS

FROM MARC

Example 28.25:

CREATE USER 'DIANE'@'localhost' IDENTIFIED BY 'SECRET'

;

CREATE VIEW NAME\_ADDRESS AS

SELECT NAME, INITIALS, STREET, HOUSENO,

TOWN

FROM PLAYERS

WHERE LEAGUENO IS NULL

;

GRANT SELECT

ON NAME\_ADDRESS

TO DIANE

Example 28.26:

CREATE USER 'GERARD'@'localhost' IDENTIFIED BY 'XYZ1234'

;

CREATE VIEW RESIDENTS (TOWN, NUMBER\_OF) AS

SELECT TOWN, COUNT(\*)

FROM PLAYERS

GROUP BY TOWN

;

GRANT SELECT

ON RESIDENTS

TO GERARD

Answer 28.1:

CREATE USER RONALDO IDENTIFIED BY 'NIKE'

Answer 28.2:

DROP USER RONALDO

Answer 28.3:

GRANT SELECT, INSERT

ON PLAYERS

TO RONALDO

Answer 28.4:

GRANT UPDATE(STREET, HOUSENO, POSTCODE, TOWN)

ON PLAYERS

TO RONALDO

Answer 28.5:

GRANT INSERT

ON TENNIS.\*

TO JACO, DIANE

Example 29.1:

SHOW INDEX FROM PLAYERS

Example 29.2:

CREATE INDEX PLAYERS\_TOWN

ON PLAYERS (TOWN)

;

SHOW INDEX FROM PLAYERS

Example 29.3:

ANALYZE TABLE PLAYERS

;

SHOW INDEX FROM PLAYERS

Example 29.4:

CHECKSUM TABLE PLAYERS

Example 29.5:

OPTIMIZE TABLE PLAYERS

Example 29.6:

CHECK TABLE PLAYERS

Example 29.7:

SELECT TABLE\_NAME, CHECK\_TIME

FROM INFORMATION\_SCHEMA.TABLES

WHERE TABLE\_NAME = 'PLAYERS'

AND TABLE\_SCHEMA = 'TENNIS'

Example 29.8:

REPAIR TABLE PLAYERS

Example 29.9:

BACKUP TABLE PLAYERS TO 'C:/WORKING\_AREA'

Example 29.10:

RESTORE TABLE PLAYERS FROM 'C:/WORKING\_AREA'

Example 31.1:

CREATE PROCEDURE DELETE\_MATCHES

(IN P\_PLAYERNO INTEGER)

BEGIN

DELETE

FROM MATCHES

WHERE PLAYERNO = P\_PLAYERNO;

END

Example 31.2:

CALL DELETE\_MATCHES (8)

Example 31.5:

CREATE PROCEDURE TEST

(OUT NUMBER1 INTEGER)

BEGIN

DECLARE NUMBER2 INTEGER DEFAULT 100;

SET NUMBER1 = NUMBER2;

END

;

CALL TEST (@NUMBER)

;

SELECT @NUMBER

Example 31.6:

CREATE PROCEDURE TEST

(OUT NUMBER1 INTEGER)

BEGIN

DECLARE NUMBER2 INTEGER

DEFAULT (SELECT COUNT(\*) FROM PLAYERS);

SET NUMBER1 = NUMBER2;

END

Example 31.7:

CREATE PROCEDURE DIFFERENCE

(IN P1 INTEGER,

IN P2 INTEGER,

OUT P3 INTEGER)

BEGIN

IF P1 > P2 THEN

SET P3 = 1;

ELSEIF P1 = P2 THEN

SET P3 = 2;

ELSE

SET P3 = 3;

END IF;

END

Example 31.8:

CREATE PROCEDURE FIBONNACI

(INOUT NUMBER1 INTEGER,

INOUT NUMBER2 INTEGER,

INOUT NUMBER3 INTEGER)

BEGIN

SET NUMBER3 = NUMBER1 + NUMBER2;

IF NUMBER3 > 10000 THEN

SET NUMBER3 = NUMBER3 - 10000;

END IF;

SET NUMBER1 = NUMBER2;

SET NUMBER2 = NUMBER3;

END

;

SET @A=16, @B=27

;

SELECT @C

;

SELECT @C

;

SELECT @C

Example 31.9:

CREATE PROCEDURE LARGEST

(OUT T CHAR(10))

BEGIN

IF (SELECT COUNT(\*) FROM PLAYERS) >

(SELECT COUNT(\*) FROM PENALTIES) THEN

SET T = 'PLAYERS';

ELSEIF (SELECT COUNT(\*) FROM PLAYERS) =

(SELECT COUNT(\*) FROM PENALTIES) THEN

SET T = 'EQUAL';

ELSE

SET T = 'PENALTIES';

END IF;

END

Example 31.10:

CREATE PROCEDURE AGE

(IN START\_DATE DATE,

IN END\_DATE DATE,

OUT YEARS INTEGER,

OUT MONTHS INTEGER,

OUT DAYS INTEGER)

BEGIN

DECLARE NEXT\_DATE, PREVIOUS\_DATE DATE;

SET YEARS = 0;

SET PREVIOUS\_DATE = START\_DATE;

SET NEXT\_DATE = START\_DATE + INTERVAL 1 YEAR;

WHILE NEXT\_DATE < END\_DATE DO

SET YEARS = YEARS + 1;

SET PREVIOUS\_DATE = NEXT\_DATE;

SET NEXT\_DATE = NEXT\_DATE + INTERVAL 1 YEAR;

END WHILE;

SET MONTHS = 0;

SET NEXT\_DATE = PREVIOUS\_DATE + INTERVAL 1 MONTH;

WHILE NEXT\_DATE < END\_DATE DO

SET MONTHS = MONTHS + 1;

SET PREVIOUS\_DATE = NEXT\_DATE;

SET NEXT\_DATE = NEXT\_DATE + INTERVAL 1 MONTH;

END WHILE;

SET DAYS = 0;

SET NEXT\_DATE = PREVIOUS\_DATE + INTERVAL 1 DAY;

WHILE NEXT\_DATE <= END\_DATE DO

SET DAYS = DAYS + 1;

SET PREVIOUS\_DATE = NEXT\_DATE;

SET NEXT\_DATE = NEXT\_DATE + INTERVAL 1 DAY;

END WHILE;

END

;

SET @START = '1991-01-12'

;

SET @END = '1999-07-09'

;

CALL AGE (@START, @END, @YEAR, @MONTH, @DAY)

;

SELECT @START, @END, @YEAR, @MONTH, @DAY

Example 31.11:

CREATE PROCEDURE SMALL\_EXIT

(OUT P1 INTEGER, OUT P2 INTEGER)

BEGIN

SET P1 = 1;

SET P2 = 1;

BLOCK1 : BEGIN

LEAVE BLOCK1;

SET P2 = 3;

END;

SET P1 = 4;

END

Example 31.12:

CREATE PROCEDURE WAIT

(IN WAIT\_SECONDS INTEGER)

BEGIN

DECLARE END\_TIME INTEGER

DEFAULT NOW() + INTERVAL WAIT\_SECONDS SECOND;

WAIT\_LOOP: LOOP

IF NOW() > END\_TIME THEN

LEAVE WAIT\_LOOP;

END IF;

END LOOP WAIT\_LOOP;

END

Example 31.13:

CREATE PROCEDURE AGAIN

(OUT RESULT INTEGER)

BEGIN

DECLARE COUNTER INTEGER DEFAULT 1;

SET RESULT = 0;

LOOP1: WHILE COUNTER <= 1000 DO

SET COUNTER = COUNTER + 1;

IF COUNTER > 100 THEN

LEAVE LOOP1;

ELSE

ITERATE LOOP1;

END IF;

SET RESULT = COUNTER \* 10;

END WHILE LOOP1;

END

Example 31.14:

CALL WAIT ((SELECT COUNT(\*) FROM PENALTIES))

Example 31.15:

CREATE TABLE PLAYERS\_WITH\_PARENTS

(PLAYERNO INTEGER NOT NULL PRIMARY KEY,

FATHER\_PLAYERNO INTEGER,

MOTHER\_PLAYERNO INTEGER)

;

ALTER TABLE PLAYERS\_WITH\_PARENTS ADD

FOREIGN KEY (FATHER\_PLAYERNO)

REFERENCES PLAYERS\_WITH\_PARENTS (PLAYERNO)

;

ALTER TABLE PLAYERS\_WITH\_PARENTS ADD

FOREIGN KEY (MOTHER\_PLAYERNO)

REFERENCES PLAYERS\_WITH\_PARENTS (PLAYERNO)

;

INSERT INTO PLAYERS\_WITH\_PARENTS VALUES

(9,NULL,NULL), (8,NULL,NULL), (7,NULL,NULL), (6,NULL,NULL),

(5,NULL,NULL), (4,8,9), (3,6,7), (2,4,5), (1,2,3)

;

CREATE PROCEDURE TOTAL\_NUMBER\_OF\_PARENTS

(IN P\_PLAYERNO INTEGER,

INOUT NUMBER INTEGER)

BEGIN

DECLARE V\_FATHER, V\_MOTHER INTEGER;

SET V\_FATHER =

(SELECT FATHER\_PLAYERNO

FROM PLAYERS\_WITH\_PARENTS

WHERE PLAYERNO = P\_PLAYERNO);

SET V\_MOTHER =

(SELECT MOTHER\_PLAYERNO

FROM PLAYERS\_WITH\_PARENTS

WHERE PLAYERNO = P\_PLAYERNO);

IF V\_FATHER IS NOT NULL THEN

CALL TOTAL\_NUMBER\_OF\_PARENTS (V\_FATHER, NUMBER);

SET NUMBER = NUMBER + 1;

END IF;

IF V\_MOTHER IS NOT NULL THEN

CALL TOTAL\_NUMBER\_OF\_PARENTS (V\_MOTHER, NUMBER);

SET NUMBER = NUMBER + 1;

END IF;

END

;

SET @NUMBER = 0

;

CALL TOTAL\_NUMBER\_OF\_PARENTS (1, @NUMBER)

;

SELECT @NUMBER

Example 31.16:

CREATE PROCEDURE TOTAL\_PENALTIES\_PLAYER

(IN P\_PLAYERNO INTEGER,

OUT TOTAL\_PENALTIES DECIMAL(8,2))

BEGIN

SELECT SUM(AMOUNT)

INTO TOTAL\_PENALTIES

FROM PENALTIES

WHERE PLAYERNO = P\_PLAYERNO;

END

;

CALL TOTAL\_PENALTIES\_PLAYER (27, @TOTAL)

;

SELECT @TOTAL

;

SELECT FATHER\_PLAYERNO, MOTHER\_PLAYERNO

INTO V\_FATHER, V\_MOTHER

FROM PLAYERS\_WITH\_PARENTS

WHERE PLAYERNO = P\_PLAYERNO

Example 31.17:

CREATE PROCEDURE GIVE\_ADDRESS

(IN P\_PLAYERNO SMALLINT,

OUT P\_STREET VARCHAR(30),

OUT P\_HOUSENO CHAR(4),

OUT P\_TOWN VARCHAR(30),

OUT P\_POSTCODE CHAR(6))

BEGIN

SELECT TOWN, STREET, HOUSENO, POSTCODE

INTO P\_TOWN, P\_STREET, P\_HOUSENO, P\_POSTCODE

FROM PLAYERS

WHERE PLAYERNO = P\_PLAYERNO;

END

Example 31.18:

CREATE TABLE FIBON

(NUMBER1 INTEGER NOT NULL PRIMARY KEY,

NUMBER2 INTEGER NOT NULL)

;

CREATE PROCEDURE FIBONNACI\_START()

BEGIN

DELETE FROM FIBON;

INSERT INTO FIBON (NUMBER, NUMBER2) VALUES (16, 27);

END

;

CREATE PROCEDURE FIBONNACI\_GIVE

(INOUT NUMBER INTEGER)

BEGIN

DECLARE N1, N2 INTEGER;

SELECT NUMBER1, NUMBER2

INTO N1, N2

FROM FIBON;

SET NUMBER = N1 + N2;

IF NUMBER > 10000 THEN

SET NUMBER = NUMBER - 10000;

END IF;

SET N1 = N2;

SET N2 = NUMBER;

UPDATE FIBON

SET NUMBER1 = N1,

NUMBER2 = N2;

END

;

CALL FIBONNACI\_START()

;

CALL FIBONNACI\_GIVE(@C)

;

SELECT @C

;

CALL FIBONNACI\_GIVE(@C)

;

SELECT @C

;

CALL FIBONNACI\_GIVE(@C)

;

SELECT @C

Example 31.19:

CREATE PROCEDURE DELETE\_PLAYER

(IN P\_PLAYERNO INTEGER)

BEGIN

DECLARE NUMBER\_OF\_ PENALTIES INTEGER;

DECLARE NUMBER\_OF\_TEAMS INTEGER;

SELECT COUNT(\*)

INTO NUMBER\_OF\_PENALTIES

FROM PENALTIES

WHERE PLAYERNO = P\_PLAYERNO;

SELECT COUNT(\*)

INTO NUMBER\_OF\_TEAMS

FROM TEAMS

WHERE PLAYERNO = P\_PLAYERNO\_;

IF NUMBER\_OF\_PENALTIES = 0 AND NUMBER\_OF\_TEAMS = 0 THEN

CALL DELETE\_MATCHES (P\_PLAYERNO);

DELETE FROM PLAYERS

WHERE PLAYERNO = P\_PLAYERNO;

END IF;

END

Example 31.20:

CREATE PROCEDURE DUPLICATE

(OUT P\_PROCESSED SMALLINT)

BEGIN

SET P\_PROCESSED = 1;

INSERT INTO TEAMS VALUES (2,27,'third');

SET P\_PROCESSED = 2;

END

;

CALL DUPLICATE(PROCESSED)

Example 31.21:

CREATE PROCEDURE SMALL\_MISTAKE1

(OUT ERROR CHAR(5))

BEGIN

DECLARE CONTINUE HANDLER FOR SQLSTATE '23000'

SET ERROR = '23000';

SET ERROR = '00000';

INSERT INTO TEAMS VALUES (2,27,'third');

END

Example 31.22:

CREATE PROCEDURE SMALL\_MISTAKE2

(OUT ERROR CHAR(5))

BEGIN

DECLARE CONTINUE HANDLER FOR SQLSTATE '23000'

SET ERROR = '23000';

DECLARE CONTINUE HANDLER FOR SQLSTATE '21S01'

SET ERROR = '21S01';

SET ERROR = '00000';

INSERT INTO TEAMS VALUES (2,27,'third',5);

END

Example 31.23:

CREATE PROCEDURE SMALL\_MISTAKE3

(OUT ERROR CHAR(5))

BEGIN

DECLARE CONTINUE HANDLER FOR SQLWARNING, NOT FOUND,

SQLEXCEPTION SET ERROR = 'XXXXX';

SET ERROR = '00000';

INSERT INTO TEAMS VALUES (2,27,'third');

END

Example 31.24:

CREATE PROCEDURE SMALL\_MISTAKE4

(OUT ERROR CHAR(5))

BEGIN

DECLARE NON\_UNIQUE CONDITION FOR SQLSTATE '23000';

DECLARE CONTINUE HANDLER FOR NON\_UNIQUE

SET ERROR = '23000';

SET ERROR = '00000';

INSERT INTO TEAMS VALUES (2,27,'third');

END

Example 31.25:

CREATE PROCEDURE SMALL\_MISTAKE5

(OUT ERROR CHAR(5))

BEGIN

DECLARE NON\_UNIQUE CONDITION FOR SQLSTATE '23000';

DECLARE CONTINUE HANDLER FOR NON\_UNIQUE

SET ERROR = '23000';

BEGIN

DECLARE CONTINUE HANDLER FOR NON\_UNIQUE

SET ERROR = '23000';

END;

BEGIN

DECLARE CONTINUE HANDLER FOR NON\_UNIQUE

SET ERROR = '00000';

INSERT INTO TEAMS VALUES (2,27,'third');

END;

END

;

CREATE PROCEDURE SMALL\_MISTAKE6 ()

BEGIN

DECLARE CONTINUE HANDLER FOR SQLSTATE '23000'

SET @PROCESSED = 100;

BEGIN

DECLARE CONTINUE HANDLER FOR SQLSTATE '23000'

SET @PROCESSED = 200;

INSERT INTO TEAMS VALUES (2,27,'third');

END;

END

Example 31.26:

CREATE PROCEDURE NUMBER\_OF\_PLAYERS

(OUT NUMBER INTEGER)

BEGIN

DECLARE A\_PLAYERNO INTEGER;

DECLARE FOUND BOOLEAN DEFAULT TRUE;

DECLARE C\_PLAYERS CURSOR FOR

SELECT PLAYERNO FROM PLAYERS;

DECLARE CONTINUE HANDLER FOR NOT FOUND

SET FOUND = FALSE;

SET NUMBER = 0;

OPEN C\_PLAYERS;

FETCH C\_PLAYERS INTO A\_PLAYERNO;

WHILE FOUND DO

SET NUMBER = NUMBER + 1;

FETCH C\_PLAYERS INTO A\_PLAYERNO;

END WHILE;

CLOSE C\_PLAYERS;

END

Example 31.27:

CREATE PROCEDURE DELETE\_OLDER\_THAN\_30()

BEGIN

DECLARE V\_AGE, V\_PLAYERNO,V\_YEARS,

V\_MONTHS, V\_DAYS INTEGER;

DECLARE V\_BIRTH\_DATE DATE;

DECLARE FOUND BOOLEAN DEFAULT TRUE;

DECLARE C\_PLAYERS CURSOR FOR

SELECT PLAYERNO, BIRTH\_DATE

FROM PLAYERS;

DECLARE CONTINUE HANDLER FOR NOT FOUND

SET FOUND = FALSE;

OPEN C\_PLAYERS;

FETCH C\_PLAYERS INTO V\_PLAYERNO, V\_BIRTH\_DATE;

WHILE FOUND DO

CALL AGE (V\_BIRTH\_DATE, NOW(), V\_YEARS,

V\_MONTHS, V\_DAYS);

IF V\_YEARS > 30 THEN

DELETE FROM PENALTIES WHERE PLAYERNO = V\_PLAYERNO;

END IF;

FETCH C\_PLAYERS INTO V\_PLAYERNO, V\_BIRTH\_DATE;

END WHILE;

CLOSE C\_PLAYERS;

END

Example 31.28:

CREATE PROCEDURE TOP\_THREE

(IN P\_PLAYERNO INTEGER,

OUT OK BOOLEAN)

BEGIN

DECLARE A\_PLAYERNO, BALANCE, SEQNO INTEGER;

DECLARE FOUND BOOLEAN;

DECLARE BALANCE\_PLAYERS CURSOR FOR

SELECT PLAYERNO, SUM(WON) - SUM(LOST)

FROM MATCHES

GROUP BY PLAYERNO

ORDER BY 2;

DECLARE CONTINUE HANDLER FOR NOT FOUND

SET FOUND = FALSE;

SET SEQNO = 0;

SET FOUND = TRUE;

SET OK = FALSE;

OPEN BALANCE\_PLAYERS;

FETCH BALANCE\_PLAYERS INTO A\_PLAYERNO, BALANCE;

WHILE FOUND AND SEQNO < 3 AND OK = FALSE DO

SET SEQNO = SEQNO + 1;

IF A\_PLAYERNO = P\_PLAYERNO THEN

SET OK = TRUE;

END IF;

FETCH BALANCE\_PLAYERS INTO A\_PLAYERNO, BALANCE;

END WHILE;

CLOSE BALANCE\_PLAYERS;

END

Example 31.29:

CREATE PROCEDURE NUMBER\_PENALTIES

(IN V\_PLAYERNO INTEGER,

OUT NUMBER INTEGER)

BEGIN

DECLARE A\_PLAYERNO INTEGER;

DECLARE FOUND BOOLEAN DEFAULT TRUE;

DECLARE C\_PLAYERS CURSOR FOR

SELECT PLAYERNO

FROM PENALTIES

WHERE PLAYERNO = V\_PLAYERNO;

DECLARE CONTINUE HANDLER FOR NOT FOUND

SET FOUND = FALSE;

SET NUMBER = 0;

OPEN C\_PLAYERS;

FETCH C\_PLAYERS INTO A\_PLAYERNO;

WHILE FOUND DO

SET NUMBER = NUMBER + 1;

FETCH C\_PLAYERS INTO A\_PLAYERNO;

END WHILE;

CLOSE C\_PLAYERS;

END

Example 31.30:

CREATE PROCEDURE ALL\_TEAMS()

BEGIN

SELECT \* FROM TEAMS;

END

;

CALL ALL\_TEAMS()

Example 31.31:

CREATE PROCEDURE NUMBERS\_OF\_ROWS()

BEGIN

SELECT COUNT(\*) FROM TEAMS;

SELECT COUNT(\*) FROM PENALTIES;

END

;

CALL NUMBER\_OF\_ROWS()

Example 31.32:

CREATE PROCEDURE USER\_VARIABLE ()

BEGIN

SET @VAR1 = 1;

END

;

CALL USER\_VARIABLE ()

;

SELECT @VAR1

Example 31.33:

CREATE DEFINER = 'CHRIS3'@'%' PROCEDURE PIPOWER

(OUT VAR1 DECIMAL(10,5))

BEGIN

SET VAR1 = POWER(PI(),2);

END

Example 31.34:

CREATE PROCEDURE POWERPOWER

(IN P1 INTEGER, OUT P2 INTEGER)

DETERMINISTIC

BEGIN

SET P2 = POWER(POWER(P1,2),2);

END

Example 31.35:

CREATE PROCEDURE CLEANUP ()

NO SQL

BEGIN

DELETE FROM PENALTIES;

END

Example 31.36:

SELECT COLUMN\_NAME

FROM INFORMATION\_SCHEMA.COLUMNS

WHERE TABLE\_SCHEMA = 'INFORMATION\_SCHEMA'

AND TABLE\_NAME = 'ROUTINES'

ORDER BY ORDINAL\_POSITION

Example 31.37:

SHOW PROCEDURE STATUS LIKE 'FIBONNACI'

Example 31.38:

SHOW CREATE PROCEDURE FIBONNACI

Example 31.39:

DROP PROCEDURE DELETE\_PLAYER

Example 31.40:

GRANT EXECUTE

ON PROCEDURE DELETE\_MATCHES

TO JOHN

Example 32.1:

CREATE FUNCTION DOLLARS(AMOUNT DECIMAL(7,2))

RETURNS DECIMAL(7,2)

BEGIN

RETURN AMOUNT \* (1 / 0.8);

END

;

SELECT PAYMENTNO, AMOUNT, DOLLARS(AMOUNT)

FROM PENALTIES

WHERE PAYMENTNO <= 3

Example 32.2:

CREATE FUNCTION NUMBER\_OF\_PLAYERS()

RETURNS INTEGER

BEGIN

RETURN (SELECT COUNT(\*) FROM PLAYERS);

END

;

SELECT NUMBER\_OF\_PLAYERS()

Example 32.3:

CREATE FUNCTION NUMBER\_OF\_PENALTIES

(P\_PLAYERNO INTEGER)

RETURNS INTEGER

BEGIN

RETURN (SELECT COUNT(\*)

FROM PENALTIES

WHERE PLAYERNO = P\_PLAYERNO);

END

;

CREATE FUNCTION NUMBER\_OF\_MATCHES

(P\_PLAYERNO INTEGER)

RETURNS INTEGER

BEGIN

RETURN (SELECT COUNT(\*)

FROM MATCHES

WHERE PLAYERNO = P\_PLAYERNO);

END

;

SELECT PLAYERNO, NAME, INITIALS

FROM PLAYERS

WHERE NUMBER\_OF\_PENALTIES(PLAYERNO) >

NUMBER\_OF\_MATCHES(PLAYERNO)

Example 32.4:

SELECT TEAMNO, DIVISION

FROM TEAMS\_NEW

WHERE DIVISION & POWER(2,3-1) = POWER(2,3-1)

;

CREATE FUNCTION POSITION\_IN\_SET

(P\_COLUMN BIGINT, POSITION SMALLINT)

RETURNS BOOLEAN

BEGIN

RETURN (P\_COLUMN & POWER(2, POSITION-1) =

POWER(2,POSITION-1));

END

;

SELECT TEAMNO, DIVISION

FROM TEAMS\_NEW

WHERE POSITION\_IN\_SET(DIVISION, 3)

Example 32.5:

CREATE FUNCTION NUMBER\_OF\_DAYS

(START\_DATE DATE,

END\_DATE DATE)

RETURNS INTEGER

BEGIN

DECLARE DAYS INTEGER;

DECLARE NEXT\_DATE, PREVIOUS\_DATE DATE;

SET DAYS = 0;

SET NEXT\_DATE = START\_DATE + INTERVAL 1 DAY;

WHILE NEXT\_DATE <= END\_DATE DO

SET DAYS = DAYS + 1;

SET PREVIOUS\_DATE = NEXT\_DATE;

SET NEXT\_DATE = NEXT\_DATE + INTERVAL 1 DAY;

END WHILE;

RETURN DAYS;

END

Example 32.6:

CREATE FUNCTION DELETE\_PLAYER

(P\_PLAYERNO INTEGER)

RETURNS BOOLEAN

BEGIN

DECLARE NUMBER\_OF\_PENALTIES INTEGER;

DECLARE NUMBER\_OF\_TEAMS INTEGER;

DECLARE EXIT HANDLER FOR SQLWARNING RETURN FALSE;

DECLARE EXIT HANDLER FOR SQLEXCEPTION RETURN FALSE;

SELECT COUNT(\*)

INTO NUMBER\_OF\_PENALTIES

FROM PENALTIES

WHERE PLAYERNO = P\_PLAYERNO;

SELECT COUNT(\*)

INTO NUMBER\_OF\_TEAMS

FROM TEAMS

WHERE PLAYERNO = P\_PLAYERNO;

IF NUMBER\_OF\_PENALTIES = 0 AND NUMBER\_OF\_TEAMS = 0 THEN

DELETE FROM MATCHES

WHERE PLAYERNO = P\_PLAYERNO;

DELETE FROM PLAYERS

WHERE PLAYERNO = P\_PLAYERNO;

END IF;

RETURN TRUE;

END

Example 32.7:

CREATE FUNCTION GET\_NUMBER\_OF\_PLAYERS()

RETURNS INTEGER

BEGIN

DECLARE NUMBER INTEGER;

CALL NUMBER\_OF\_PLAYERS(NUMBER);

RETURN NUMBER;

END

Example 32.8:

CREATE FUNCTION OVERLAP\_BETWEEN\_PERIODS

(PERIOD1\_START DATETIME,

PERIOD1\_END DATETIME,

PERIOD2\_START DATETIME,

PERIOD2\_END DATETIME)

RETURNS BOOLEAN

BEGIN

DECLARE TEMPORARY\_DATE DATETIME;

IF PERIOD1\_START > PERIOD1\_END THEN

SET TEMPORARY\_DATE = PERIOD1\_START;

SET PERIOD1\_START = PERIOD1\_END;

SET PERIOD1\_END = TEMPORARY\_DATE;

END IF;

IF PERIOD2\_START > PERIOD2\_END THEN

SET TEMPORARY\_DATE = PERIOD2\_START;

SET PERIOD2\_START = PERIOD2\_END;

SET PERIOD2\_END = TEMPORARY\_DATE;

END IF;

RETURN NOT(PERIOD1\_END < PERIOD2\_START OR

PERIOD2\_END < PERIOD1\_START);

END

Example 32.9:

SELECT \*

FROM COMMITTEE\_MEMBERS

WHERE OVERLAP\_BETWEEN\_PERIODS(BEGIN\_DATE,END\_DATE,

'1991-06-30','1992-06-30')

ORDER BY 1, 2

Example 32.10:

DROP FUNCTION PLACE\_IN\_SET

Example 33.1:

CREATE TABLE CHANGES

(USER CHAR(30) NOT NULL,

CHA\_TIME TIMESTAMP NOT NULL,

CHA\_PLAYERNO SMALLINT NOT NULL,

CHA\_TYPE CHAR(1) NOT NULL,

CHA\_PLAYERNO\_NEW INTEGER,

PRIMARY KEY (USER, CHA\_TIME,

CHA\_PLAYERNO, CHA\_TYPE))

Example 33.2:

CREATE TRIGGER INSERT\_PLAYERS

AFTER

INSERT ON PLAYERS FOR EACH ROW

BEGIN

INSERT INTO CHANGES

(USER, CHA\_TIME, CHA\_PLAYERNO,

CHA\_TYPE, CHA\_PLAYERNO\_NEW)

VALUES (USER, CURDATE(), NEW.PLAYERNO, 'I', NULL);

END

;

CREATE PROCEDURE INSERT\_CHANGE

(IN CPNO INTEGER,

IN CTYPE CHAR(1),

IN CPNO\_NEW INTEGER)

BEGIN

INSERT INTO CHANGES (USER, CHA\_TIME, CHA\_PLAYERNO,

CHA\_TYPE, CHA\_PLAYERNO\_NEW)

VALUES (USER, CURDATE(), CPNO, CTYPE, CPNO\_NEW);

END

;

CREATE TRIGGER INSERT\_PLAYER

AFTER INSERT ON PLAYERS FOR EACH ROW

BEGIN

CALL INSERT\_CHANGE(NEW.PLAYERNO, 'I', NULL);

END

Example 33.3:

CREATE TRIGGER DELETE\_PLAYER

AFTER DELETE ON PLAYERS FOR EACH ROW

BEGIN

CALL INSERT\_CHANGE (OLD.PLAYERNO, 'D', NULL);

END

Example 33.4:

CREATE TRIGGER UPDATE\_PLAYER

AFTER UPDATE ON PLAYERS FOR EACH ROW

BEGIN

CALL INSERT\_CHANGES

(NEW.PLAYERNO, 'U', OLD.PLAYERNO);

END

Example 33.5:

CREATE TABLE PLAYERS\_MAT

(PLAYERNO INTEGER NOT NULL PRIMARY KEY,

NUMBER\_OF\_MATCHES INTEGER NOT NULL)

;

INSERT INTO PLAYERS\_MAT (PLAYERNO, NUMBER\_OF\_MATCHES)

SELECT PLAYERNO,

(SELECT COUNT(\*)

FROM MATCHES AS M

WHERE P.PLAYERNO = M.PLAYERNO)

FROM PLAYERS AS P

Example 33.6:

CREATE TRIGGER INSERT\_PLAYERS

AFTER INSERT ON PLAYERS FOR EACH ROW

BEGIN

INSERT INTO PLAYERS\_MAT

VALUES(NEW.PLAYERNO, 0);

END

Example 33.7:

CREATE TRIGGER DELETE\_PLAYERS

AFTER DELETE ON PLAYERS FOR EACH ROW

BEGIN

DELETE FROM PLAYERS\_MAT

WHERE PLAYERNO = OLD.PLAYERNO;

END

Example 33.8:

CREATE TRIGGER INSERT\_MATCHES

AFTER INSERT ON MATCHES FOR EACH ROW

BEGIN

UPDATE PLAYERS\_MAT

SET NUMBER\_OF\_MATCHES = NUMBER\_OF\_MATCHES + 1

WHERE PLAYERNO = NEW.PLAYERNO;

END

Example 33.9:

CREATE TRIGGER DELETE\_MATCHES

AFTER DELETE ON MATCHES FOR EACH ROW

BEGIN

UPDATE PLAYERS\_MAT

SET NUMBER\_OF\_MATCHES = NUMBER\_OF\_MATCHES - 1

WHERE PLAYERNO = OLD.PLAYERNO;

END

Example 33.10:

CREATE TRIGGER SUM\_PENALTIES\_INSERT

AFTER INSERT ON PENALTIES FOR EACH ROW

BEGIN

DECLARE TOTAL DECIMAL(8,2);

SELECT SUM(AMOUNT)

INTO TOTAL

FROM PENALTIES

WHERE PLAYERNO = NEW.PLAYERNO;

UPDATE PLAYERS

SET SUM\_PENALTIES = TOTAL

WHERE PLAYERNO = NEW.PLAYERNO

END

;

CREATE TRIGGER SUM\_PENALTIES\_DELETE

AFTER DELETE, UPDATE ON PENALTIES FOR EACH ROW

BEGIN

DECLARE TOTAL DECIMAL(8,2);

SELECT SUM(AMOUNT)

INTO TOTAL

FROM PENALTIES

WHERE PLAYERNO = OLD.PLAYERNO;

UPDATE PLAYERS

SET SUM\_PENALTIES = TOTAL

WHERE PLAYERNO = OLD.PLAYERNO

END

;

UPDATE PLAYERS

SET SUM\_PENALTIES = (SELECT SUM(AMOUNT)

FROM PENALTIES

WHERE PLAYERNO = NEW.PLAYERNO)

WHERE PLAYERNO = NEW.PLAYERNO

Example 33.11:

CREATE TRIGGER BORN\_VS\_JOINED

BEFORE INSERT, UPDATE ON PLAYERS FOR EACH ROW

BEGIN

IF YEAR(NEW.BIRTH\_DATE) >= NEW.JOINED) THEN

ROLLBACK WORK;

END IF;

END

Example 33.12:

CREATE TRIGGER FOREIGN\_KEY1

BEFORE INSERT, UPDATE ON PENALTIES FOR EACH ROW

BEGIN

IF (SELECT COUNT(\*) FROM PLAYERS

WHERE PLAYERNO = NEW.PLAYERNO) = 0 THEN

ROLLBACK WORK;

END IF;

END

;

CREATE TRIGGER FOREIGN\_KEY2

BEFORE DELETE, UPDATE ON PLAYERS FOR EACH ROW

BEGIN

DELETE

FROM PENALTIES

WHERE PLAYERNO = OLD.PLAYERNO;

END

Example 33.13:

DROP TRIGGER BORN\_VS\_JOINED

Answer 33.2:

CREATE TRIGGER MAX1

AFTER INSERT, UPDATE(POSITION) OF COMMITTEE\_MEMBERS

FOR EACH ROW

BEGIN

SELECT COUNT(\*)

INTO NUMBER\_MEMBERS

FROM COMMITTEE\_MEMBERS

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM COMMITTEE\_MEMBERS

WHERE CURRENT DATE BETWEEN

BEGIN\_DATE AND END\_DATE

GROUP BY POSITION

HAVING COUNT(\*) > 1)

IF NUMBER\_MEMBERS > 0 THEN

ROLLBACK WORK;

ENDIF;

END

Answer 33.3:

CREATE TRIGGER SUM\_PENALTIES\_250

AFTER INSERT, UPDATE(AMOUNT) OF PENALTIES

FOR EACH ROW

BEGIN

SELECT COUNT(\*)

INTO NUMBER\_PENALTIES

FROM PENALTIES

WHERE PLAYERNO IN

(SELECT PLAYERNO

FROM PENALTIES

GROUP BY PLAYERNO

HAVING SUM(AMOUNT) > 250);

IF NUMBER\_PENALTIES > 0 THEN

ROLLBACK WORK;

ENDIF;

END

Answer 33.4:

CREATE TRIGGER NUMBER\_MATCHES\_INSERT

AFTER INSERT OF MATCHES FOR EACH ROW

BEGIN

UPDATE TEAMS

SET NUMBER\_MATCHES =

(SELECT COUNT(\*)

FROM MATCHES

WHERE PLAYERNO = NEW.PLAYERNO)

WHERE PLAYERNO = NEW.PLAYERNO

END

;

CREATE TRIGGER NUMBER\_MATCHES\_DELETE

AFTER DELETE, UPDATE OF MATCHES FOR EACH ROW

BEGIN

UPDATE TEAMS

SET NUMBER\_MATCHES =

(SELECT COUNT(\*)

FROM MATCHES

WHERE PLAYERNO = OLD.PLAYERNO)

WHERE PLAYERNO = OLD.PLAYERNO

END

Example 34.1:

SET GLOBAL EVENT\_SCHEDULER = TRUE

;

SET GLOBAL EVENT\_SCHEDULER = FALSE

;

CREATE TABLE EVENTS\_INVOKED

(EVENT\_NAME VARCHAR(20) NOT NULL,

EVENT\_STARTED TIMESTAMP NOT NULL)

Example 34.2:

CREATE EVENT DIRECT

ON SCHEDULE AT NOW()

DO INSERT INTO EVENTS\_INVOKED VALUES ('DIRECT', NOW())

;

SELECT \*

FROM EVENTS\_INVOKED

WHERE EVENT\_NAME = 'DIRECT'

;

CREATE EVENT TENNIS.DIRECT

ON SCHEDULE AT NOW()

DO INSERT INTO EVENTS\_INVOKED VALUES ('DIRECT', NOW())

Example 34.3:

CREATE EVENT END2010

ON SCHEDULE AT '2010-12-31 11:00:00'

DO INSERT INTO EVENTS\_INVOKED VALUES ('END2010', NOW())

Example 34.4:

CREATE EVENT THREEDAYS

ON SCHEDULE AT NOW() + INTERVAL 3 DAY

DO INSERT INTO EVENTS\_INVOKED VALUES ('THREEDAYS', NOW())

Example 34.5:

CREATE EVENT NEXT\_SUNDAY

ON SCHEDULE AT

CASE DAYNAME(NOW())

WHEN 'Sunday' THEN NOW() + INTERVAL 7 DAY

WHEN 'Monday' THEN NOW() + INTERVAL 6 DAY

WHEN 'Tuesday' THEN NOW() + INTERVAL 5 DAY

WHEN 'Wednesday' THEN NOW() + INTERVAL 4 DAY

WHEN 'Thursday' THEN NOW() + INTERVAL 3 DAY

WHEN 'Friday' THEN NOW() + INTERVAL 2 DAY

WHEN 'Saturday' THEN NOW() + INTERVAL 1 DAY

END

DO INSERT INTO EVENTS\_INVOKED

VALUES ('NEXT\_SUNDAY',NOW())

;

CREATE EVENT NEXT\_SUNDAY

ON SCHEDULE AT

NOW() + INTERVAL (8 - DAYOFWEEK(NOW())) DAY

DO INSERT INTO EVENTS\_INVOKED

VALUES ('NEXT\_SUNDAY',NOW())

Example 34.6:

CREATE EVENT MORNING11

ON SCHEDULE AT TIMESTAMP(CURDATE() +

INTERVAL 1 DAY, '11:00:00')

DO INSERT INTO EVENTS\_INVOKED VALUES ('MORNING11', NOW())

Example 34.7:

CREATE EVENT EVERY2HOUR

ON SCHEDULE EVERY 2 HOUR

STARTS NOW() + INTERVAL 3 HOUR

ENDS CURDATE() + INTERVAL 23 HOUR

DO INSERT INTO EVENTS\_INVOKED VALUES ('EVERY2HOUR', NOW())

Example 34.8:

CREATE EVENT SIXTIMES

ON SCHEDULE EVERY 1 MINUTE

STARTS TIMESTAMP(CURDATE() + INTERVAL 1 DAY,'12:00:00')

ENDS TIMESTAMP(CURDATE() + INTERVAL 1 DAY,'12:00:00')

+ INTERVAL 5 MINUTE

DO INSERT INTO EVENTS\_INVOKED

VALUES ('SIXTIMES', NOW())

Example 34.9:

CREATE EVENT FIVESUNDAYS

ON SCHEDULE EVERY 1 WEEK

STARTS CASE DAYNAME(NOW())

WHEN 'Sunday' THEN NOW()

WHEN 'Monday' THEN NOW() + INTERVAL 6 DAY

WHEN 'Tuesday' THEN NOW() + INTERVAL 5 DAY

WHEN 'Wednesday' THEN NOW() + INTERVAL 4 DAY

WHEN 'Thursday' THEN NOW() + INTERVAL 3 DAY

WHEN 'Friday' THEN NOW() + INTERVAL 2 DAY

WHEN 'Saturday' THEN NOW() + INTERVAL 1 DAY

END

ENDS CASE DAYNAME(NOW())

WHEN 'Sunday' THEN NOW()

WHEN 'Monday' THEN NOW() + INTERVAL 6 DAY

WHEN 'Tuesday' THEN NOW() + INTERVAL 5 DAY

WHEN 'Wednesday' THEN NOW() + INTERVAL 4 DAY

WHEN 'Thursday' THEN NOW() + INTERVAL 3 DAY

WHEN 'Friday' THEN NOW() + INTERVAL 2 DAY

WHEN 'Saturday' THEN NOW() + INTERVAL 1 DAY

END + INTERVAL 4 WEEK

DO INSERT INTO EVENTS\_INVOKED

VALUES ('FIVESUNDAYS',NOW())

Example 34.10:

CREATE EVENT SUNDAYS

ON SCHEDULE EVERY 1 WEEK

STARTS TIMESTAMP(CASE DAYNAME(NOW())

WHEN 'Sunday' THEN NOW()

WHEN 'Monday' THEN NOW() + INTERVAL 6 DAY

WHEN 'Tuesday' THEN NOW() + INTERVAL 5 DAY

WHEN 'Wednesday' THEN NOW() + INTERVAL 4 DAY

WHEN 'Thursday' THEN NOW() + INTERVAL 3 DAY

WHEN 'Friday' THEN NOW() + INTERVAL 2 DAY

WHEN 'Saturday' THEN NOW() + INTERVAL 1 DAY

END, '15:00:00')

ENDS TIMESTAMP(

CASE DAYNAME(CONCAT(YEAR(CURDATE()),'-12-31'))

WHEN 'Sunday' THEN

CONCAT(YEAR(CURDATE()),'-12-31')

WHEN 'Monday' THEN

CONCAT(YEAR(CURDATE()),'-12-31') - INTERVAL 1 DAY

WHEN 'Tuesday' THEN

CONCAT(YEAR(CURDATE()),'-12-31') - INTERVAL 2 DAY

WHEN 'Wednesday' THEN

CONCAT(YEAR(CURDATE()),'-12-31') - INTERVAL 3 DAY

WHEN 'Thursday' THEN

CONCAT(YEAR(CURDATE()),'-12-31') - INTERVAL 4 DAY

WHEN 'Friday' THEN

CONCAT(YEAR(CURDATE()),'-12-31') - INTERVAL 5 DAY

WHEN 'Saturday' THEN

CONCAT(YEAR(CURDATE()),'-12-31') - INTERVAL 6 DAY

END, '15:00:00')

DO INSERT INTO EVENTS\_INVOKED VALUES ('SUNDAYS', NOW())

Example 34.11:

CREATE EVENT STARTMONTH

ON SCHEDULE EVERY 1 MONTH

STARTS CURDATE() + INTERVAL 1 MONTH -

INTERVAL (DAYOFMONTH(CURDATE()) - 1) DAY

ENDS TIMESTAMP(CONCAT(YEAR(CURDATE()),'-12-31'))

DO INSERT INTO EVENTS\_INVOKED

VALUES ('STARTMONTH', NOW())

Example 34.12:

CREATE EVENT QUARTERS

ON SCHEDULE EVERY 3 MONTH

STARTS (CURDATE() - INTERVAL (DAYOFMONTH(CURDATE())

- 1) DAY) - INTERVAL (MOD(MONTH(CURDATE()

- INTERVAL (DAYOFMONTH(CURDATE()) - 1) DAY)+2,3)) MONTH

+ INTERVAL 3 MONTH

DO INSERT INTO EVENTS\_INVOKED VALUES ('QUARTERS', NOW())

Example 34.13:

CREATE EVENT END\_OF\_YEAR

ON SCHEDULE EVERY 1 YEAR

STARTS ((NOW() - INTERVAL (DAYOFYEAR(NOW()) - 1) DAY)

+ INTERVAL 1 YEAR)

- INTERVAL 1 DAY

ENDS '2025-12-31'

DO INSERT INTO EVENTS\_INVOKED VALUES ('END\_OF\_YEAR', NOW())

Example 34.14:

CREATE EVENT NOT2020

ON SCHEDULE EVERY 1 YEAR

STARTS ((NOW() - INTERVAL (DAYOFYEAR(NOW()) - 1) DAY)

+ INTERVAL 1 YEAR)

- INTERVAL 1 DAY

ENDS '2025-12-31'

DO BEGIN

IF YEAR(CURDATE()) <> 2020 THEN

INSERT INTO EVENTS\_INVOKED

VALUES ('NOT2020', NOW());

END IF;

END

Example 34.15:

CREATE TABLE MATCHES\_ANNUALREPORT

(PLAYERNO INTEGER NOT NULL,

YEAR INTEGER NOT NULL,

NUMBER INTEGER NOT NULL,

PRIMARY KEY (PLAYERNO, YEAR),

FOREIGN KEY (PLAYERNO) REFERENCES PLAYERS (PLAYERNO))

Example 34.16:

CREATE EVENT YEARBALANCING

ON SCHEDULE EVERY 1 YEAR

STARTS ((NOW() - INTERVAL (DAYOFYEAR(NOW()) - 1) DAY)

+ INTERVAL 1 YEAR)

- INTERVAL 1 DAY

DO INSERT INTO MATCHES\_ANNUALREPORT

SELECT PLAYERNO, YEAR, COUNT(\*)

FROM MATCHES

WHERE YEAR(DATE) = YEAR(CURDATE())

GROUP BY PLAYERNO, YEAR

Example 34.17:

CREATE EVENT DIRECT

ON SCHEDULE AT NOW()

ON COMPLETION PRESERVE

DO INSERT INTO EVENTS\_INVOKED VALUES ('DIRECT', NOW())

Example 34.18:

CREATE EVENT DIRECT\_WITH\_COMMENT

ON SCHEDULE AT NOW()

ON COMPLETION PRESERVE

COMMENT 'This event starts directly'

DO INSERT INTO EVENTS\_INVOKED

VALUES ('DIRECT\_WITH\_COMMENT', NOW())

Example 34.19:

CREATE EVENT DIRECT\_INACTIVE

ON SCHEDULE AT NOW()

ON COMPLETION PRESERVE

DISABLE

COMMENT 'This event is inactive'

DO INSERT INTO EVENTS\_INVOKED

VALUES ('DIRECT\_INACTIVE', NOW())

Example 34.20:

ALTER EVENT STARTMONTH

ON SCHEDULE EVERY 1 MONTH

STARTS CURDATE() + INTERVAL 1 MONTH -

INTERVAL (DAYOFMONTH(CURDATE()) - 1) DAY

ENDS TIMESTAMP('2025-12-31')

Example 34.21:

ALTER EVENT STARTMONTH

RENAME TO FIRST\_OF\_THE\_MONTH

Example 34.22:

ALTER EVENT DIRECT\_INACTIVE

ENABLE

Example 34.23:

DROP EVENT FIRST\_OF\_THE\_MONTH

Example 34.24:

GRANT EVENT

ON TENNIS.\*

TO SAM

Example 34.25:

SHOW CREATE EVENT TOMORROW11

Example 35.1:

<HTML>

<HEAD>

<TITLE>Logging on</TITLE>

</HEAD>

<BODY>

<?php

$host = "localhost";

$user = "root";

$pass = "root";

$conn = mysql\_connect($host, $user, $pass)

or die ("<p>Logging on has not succeeded.</p>");

echo "<p>Logging on has succeeded.</p>\n";

mysql\_close($conn);

?>

</BODY>

</HTML>

Example 35.2:

<HTML>

<HEAD>

<TITLE>Current database</TITLE>

</HEAD>

<BODY>

<?php

$host = "localhost";

$user = "root";

$pass = "root";

$conn = mysql\_connect($host, $user, $pass)

or die ("<p>Logging on has not succeeded.\n");

echo "<p>Logging on has succeeded.\n";

$db = mysql\_select\_db("TENNIS")

or die ("<br>Database unknown.\n");

echo "<br>TENNIS is the current database now.\n";

mysql\_close($conn);

?>

</BODY>

</HTML>

Example 35.3:

<HTML>

<HEAD>

<TITLE>Create Index</TITLE>

</HEAD>

<BODY>

<?php

$host = "localhost";

$user = "root";

$pass = "root";

$conn = mysql\_connect($host, $user, $pass)

or die ("<p>Logging on has not succeeded.\n");

echo "<p>Logging on has succeeded.\n";

$db = mysql\_select\_db("TENNIS")

or die ("<br>Database unknown.\n");

echo "<br>TENNIS is the current database now.\n";

$result = mysql\_query("CREATE UNIQUE INDEX PLAY

ON PLAYERS (PLAYERNO)");

if (!$result)

{

echo "<br>Index PLAY is not created!\n";

}

else

{

echo "<br>Index PLAY is created!\n";

};

mysql\_close($conn);

?>

</BODY>

</HTML>

Example 35.4:

<HTML>

<HEAD>

<TITLE>Create Index plus response</TITLE>

</HEAD>

<BODY>

<?php

$host = "localhost";

$user = "root";

$pass = "root";

$conn = mysql\_connect($host, $user, $pass)

or die ("<p>Logging on has not succeeded.\n");

echo "<p>Logging on has succeeded.\n";

$db = mysql\_select\_db("TENNIS")

or die ("<br>Database unknown.\n");

echo "<br>TENNIS is the current database now.\n";

$result = mysql\_query("CREATE UNIQUE INDEX PLAY

ON PLAYERS (PLAYERNO)");

if (!$result)

{

echo "<br>Index PLAY is not created!\n";

}

else

{

echo "<br>Index PLAY is created!\n";

};

echo "<br>mysql\_info=".mysql\_info($conn);

mysql\_close($conn);

?>

</BODY>

</HTML>

Example 35.5:

<HTML>

<HEAD>

<TITLE>Error messages</TITLE>

</HEAD>

<BODY>

<?php

$host = "localhost";

$user = "root";

$pass = "root";

$conn = mysql\_connect($host, $user, $pass)

or die ("<p>Logging on has not succeeded.\n");

echo "<p>Logging on has succeeded.\n";

$db = mysql\_select\_db("TENNIS")

or die ("<br>Database unknown.\n");

echo "<br>TENNIS is the current database now.\n";

$result = mysql\_query("CREATE UNIQUE INDEX PLAY

ON PLAYERS (PLAYERNO)");

if (!$result)

{

echo "<br>Index PLAY is not created!\n";

$error\_number = mysql\_errno();

$error\_message = mysql\_error();

echo "<br>Fout: $error\_number: $error\_message\n";

}

else

{

echo "<br>Index PLAY is created!\n";

}

mysql\_close($conn);

?>

</BODY>

</HTML>

Example 35.6:

<HTML>

<HEAD>

<TITLE>Two connections</TITLE>

</HEAD>

<BODY>

<?php

$host = "localhost";

$user = "root";

$pass = "root";

$conn1 = mysql\_connect($host, $user, $pass)

or die ("<p>Logging on has not succeeded.\n");

echo "<p>Logging on has succeeded.\n";

$host = "localhost";

$user = "BOOKSQL";

$pass = "BOOKSQLPW";

$conn2 = mysql\_connect($host, $user, $pass)

or die ("<p>Logging on has not succeeded.\n");

echo "<p>Logging on has succeeded.\n";

$db = mysql\_select\_db("TENNIS", $conn1)

or die ("<br>Database unknown.\n");

echo "<br>Connection 1 is started.\n";

$db = mysql\_select\_db("TENNIS", $conn2)

or die ("<br>Database unknown.\n");

echo "<br>Connection 2 is started.\n";

mysql\_close($conn1);

mysql\_close($conn2);

?>

</BODY>

</HTML>

Example 35.7:

<HTML>

<HEAD>

<TITLE>Parameters</TITLE>

</HEAD>

<BODY>

<?php

$host = "localhost";

$user = "root";

$pass = "root";

$conn = mysql\_connect($host, $user, $pass)

or die ("<p>Logging on has not succeeded.\n");

echo "<p>Logging on has succeeded.\n";

$db = mysql\_select\_db("TENNIS")

or die ("<br>Database unknown.\n");

echo "<br>TENNIS is the current database now.\n";

$wnr = 22;

$result = mysql\_query("UPDATE MATCHES

SET WON = WON + 1 WHERE MATCHNO = $mno");

if (!$result)

{

echo "<br>Update not executed!\n";

$error\_number = mysql\_errno();

$error\_message = mysql\_error();

echo "<br>Error: $error\_number: $error\_message\n";

}

else

{

echo "<br>WON column has increased for match $mno.\n";

}

mysql\_close($conn);

?>

</BODY>

</HTML>

Example 35.8:

<HTML>

<HEAD>

<TITLE>Query with a row</TITLE>

</HEAD>

<BODY>

<?php

$host = "localhost";

$user = "root";

$pass = "root";

$conn = mysql\_connect($host, $user, $pass)

or die ("<p>Logging on has not succeeded.\n");

echo "<p>Logging on has succeeded.\n";

$db = mysql\_select\_db("TENNIS")

or die ("<br>Database unknown.\n");

echo "<br>TENNIS is the current database now.\n";

$query = "SELECT COUNT(\*) AS NUMBER FROM PLAYERS";

$result = mysql\_query($query)

or die ("<br>Query is incorrect.\n");

$row = mysql\_fetch\_assoc($result)

or die ("<br>Query had no result.\n");

echo "<br>The number of players ".$row['NUMBER'].".\n";

mysql\_close($conn);

?>

</BODY>

</HTML>

Example 35.9:

<HTML>

<HEAD>

<TITLE>SELECT statement with multiple rows</TITLE>

</HEAD>

<BODY>

<?php

$host = "localhost";

$user = "root";

$pass = "root";

$conn = mysql\_connect($host, $user, $pass)

or die ("<p>Logging on has not succeeded.\n");

echo "<p>Logging on has not succeeded.\n";

$db = mysql\_select\_db("TENNIS")

or die ("<br>Database unknown.\n");

echo "<br>TENNIS is the current database now.\n";

$query = "SELECT PLAYERNO FROM PLAYERS ORDER BY 1 DESC";

$result = mysql\_query($query)

or die ("<br>Query is incorrect.\n");

if (mysql\_num\_rows($result) > 0)

{

while ($row=mysql\_fetch\_assoc($result))

{

echo "<br>Player number ".$row['PLAYERNO'].".\n";

}

}

else

{

echo "<br>No players found.\n";

}

mysql\_free\_result($result);

mysql\_close($conn);

?>

</BODY>

</HTML>

Example 35.10:

<HTML>

<HEAD>

<TITLE>MYSQL\_FETCH\_ROW function</TITLE>

</HEAD>

<BODY>

<?php

$host = "localhost";

$user = "root";

$pass = "root";

$conn = mysql\_connect($host, $user, $pass)

or die ("<p>Logging on has not succeeded.\n");

echo "<p>Logging on has succeeded.\n";

$db = mysql\_select\_db("TENNIS")

or die ("<br>Database unknown.\n");

echo "<br>TENNIS is the current database now.\n";

$query = "SELECT PLAYERNO FROM PLAYERS ORDER BY 1 DESC";

$result = mysql\_query($query)

or die ("<br>Query is incorrect.\n");

while ($row=mysql\_fetch\_row($result))

{

echo "<br>Player number ".$row[0].".\n";

};

mysql\_free\_result($result);

mysql\_close($conn);

?>

</BODY>

</HTML>

Example 35.11:

<HTML>

<HEAD>

<TITLE>MYSQL\_DATA\_SEEK function</TITLE></HEAD>

<BODY>

<?php

$host = "localhost";

$user = "root";

$pass = "root";

$conn = mysql\_connect($host, $user, $pass)

or die ("<p>Logging on has not succeeded.\n");

echo "<p>Logging on has succeeded.\n";

$db = mysql\_select\_db("TENNIS")

or die ("<br>Database unknown.\n");

echo "<br>TENNIS is the current database now.\n";

$query = "SELECT PLAYERNO FROM PLAYERS ORDER BY 1 DESC";

$result = mysql\_query($query)

or die ("<br>Query is incorrect.\n");

mysql\_data\_seek($result, 3);

$row=mysql\_fetch\_row($result);

echo "<br>Player number ".$row[0].".\n";

mysql\_close($conn);

?>

</BODY>

</HTML>

Example 35.12:

<HTML>

<HEAD>

<TITLE>Working with objects</TITLE>

</HEAD>

<BODY>

<?php

$host = "localhost";

$user = "root";

$pass = "root";

$conn = mysql\_connect($host, $user, $pass)

or die ("<p>Logging on has not succeeded.\n");

echo "<p>Logging on has succeeded.\n";

$db = mysql\_select\_db("TENNIS")

or die ("<br>Database unknown.\n");

echo "<br>TENNIS is the current database now.\n";

$query = "SELECT PLAYERNO FROM PLAYERS ORDER BY 1 DESC";

$result = mysql\_query($query)

or die ("<br>Query is incorrect.\n");

while ($row=mysql\_fetch\_object($result))

{

echo "<br>Player number ".$row->PLAYERNO.".\n";

};

mysql\_free\_result($result);

mysql\_close($conn);

?>

</BODY>

</HTML>

Example 35.13:

<HTML>

<HEAD>

<TITLE>Query with null values</TITLE>

</HEAD>

<BODY>

<?php

$host = "localhost";

$user = "root";

$pass = "root";

$conn = mysql\_connect($host, $user, $pass)

or die ("<p>Logging on has not succeeded.\n");

echo "<p>Logging on has succeeded.\n";

$db = mysql\_select\_db("TENNIS")

or die ("<br>Database unknown.\n");

echo "<br>TENNIS is the current database now.\n";

$query = "SELECT LEAGUENO FROM PLAYERS";

$result = mysql\_query($query)

or die ("<br>Query is incorrect.\n");

if (mysql\_num\_rows($result) > 0)

{

while ($row=mysql\_fetch\_assoc($result))

{

if ($row['LEAGUENO'] === NULL)

{

echo "<br>Player number is unknown.\n";

}

else

{

echo "<br>Player number ".$row['LEAGUENO'].".\n";

}

}

}

else

{

echo "<br>No players found.\n";

}

mysql\_close($conn);

?>

</BODY>

</HTML>

Example 35.14:

<HTML>

<HEAD>

<TITLE>Characteristics of expressions</TITLE>

</HEAD>

<BODY>

<?php

$host = "localhost";

$user = "root";

$pass = "root";

$conn = mysql\_connect($host, $user, $pass)

or die ("<p>Logging on has not succeeded.\n");

echo "<p>Logging on has succeeded.\n";

$db = mysql\_select\_db("TENNIS")

or die ("<br>Database unknown.\n");

echo "<br>TENNIS is the current database now.\n";

$query = "SELECT \* FROM PLAYERS WHERE PLAYERNO = 27";

$result = mysql\_query($query)

or die ("<br>Query is incorrect.\n");

while ($field=mysql\_fetch\_field($result))

{

echo "<br>".$field->name." ".$field->type." ".

$field->max\_length." ".$field->primary\_key."\n";

}

mysql\_close($conn);

?>

</BODY>

</HTML>

Example 35.15:

<HTML>

<HEAD>

<TITLE>Characteristics of expressions</TITLE>

</HEAD>

<BODY>

<?php

$host = "localhost";

$user = "root";

$pass = "root";

$conn = mysql\_connect($host, $user, $pass)

or die ("<p>Logging on has not succeeded.\n");

echo "<p>Logging on has succeeded.\n";

$db = mysql\_select\_db("TENNIS")

or die ("<br>Database unknown.\n");

echo "<br>TENNIS is the current database now.\n";

$query = "SELECT \* FROM PLAYERS WHERE PLAYERNO = 27";

$result = mysql\_query($query)

or die ("<br>Query is incorrect.\n");

$exp = 0;

while ($field=mysql\_fetch\_field($result))

{

echo "<br>Name=".mysql\_field\_name($result, $exp)."\n";

echo "<br>Data type=".mysql\_field\_type($result, $exp)."\n";

echo "<br>Length=".mysql\_field\_len($result, $exp)."\n";

echo "<br>Table=".mysql\_field\_table($result, $exp)."\n";

$exp += 1;

}

mysql\_close($conn);

?>

</BODY>

</HTML>

Example 35.16:

<HTML>

<HEAD>

<TITLE>Catalog tables</TITLE>

</HEAD>

<BODY>

<?php

$host = "localhost";

$user = "root";

$pass = "root";

$conn = mysql\_connect($host, $user, $pass)

or die ("<p>Logging on has not succeeded.\n");

echo "<p>Logging on has succeeded.\n";

$db = mysql\_select\_db("TENNIS")

or die ("<br>Database unknown.\n");

echo "<br>TENNIS is the current database now.\n";

$query1 = "SELECT TABLE\_NAME, COLUMN\_NAME

FROM INFORMATION\_SCHEMA.COLUMNS

WHERE TABLE\_NAME IN

('COMMITTEE\_MEMBERS','PENALTIES','PLAYERS',

'TEAMS','MATCHES')

ORDER BY TABLE\_NAME, ORDINAL\_POSITION";

$tables = mysql\_query($query1)

or die ("<br>Query1 is incorrect.\n");

while ($tablerow=mysql\_fetch\_assoc($tables))

{

$query2 = "SELECT COUNT(DISTINCT ";

$query2 .= $tablerow['COLUMN\_NAME'].") AS A, ";

$query2 .= "MIN( ".$tablerow['COLUMN\_NAME'].") AS B, ";

$query2 .= "MAX( ".$tablerow['COLUMN\_NAME'].") AS C ";

$query2 .= "FROM ".$tablerow['TABLE\_NAME'];

$columns = mysql\_query($query2)

or die ("<br>Query2 is incorrect.\n");

$columnrow=mysql\_fetch\_assoc($columns);

echo "<br>".$tablerow['TABLE\_NAME'].".".

$tablerow['COLUMN\_NAME'].

" Different=".$columnrow['A'].

" Minimum=".$columnrow['B'].

" Maximum=".$columnrow['C']."\n";

mysql\_free\_result($columns);

};

mysql\_free\_result($tables);

mysql\_close($conn);

?>

</BODY>

</HTML>

Example 36.1:

PREPARE S1 FROM 'SELECT \* FROM TEAMS'

;

EXECUTE S1

;

SET @SQL\_STATEMENT = 'SELECT \* FROM TEAMS'

;

PREPARE S1 FROM @SQL\_STATEMENT

Example 36.2:

DEALLOCATE PREPARE S1

Example 36.3:

PREPARE S2 FROM 'SELECT \* FROM TEAMS WHERE TEAMNO = @TNO'

;

SET @TNO = 1

;

EXECUTE S2

;

SET @TNO = 2

;

EXECUTE S2

Example 36.4:

PREPARE S3 FROM

'SELECT \* FROM TEAMS WHERE TEAMNO BETWEEN ? AND ?'

;

SET @FROM\_TNO = 1, @TO\_TNO = 4

;

EXECUTE S3 USING @FROM\_TNO, @TO\_TNO

;

DEALLOCATE PREPARE S3

Example 36.5:

CREATE PROCEDURE DROP\_TABLE

(IN TABLENAME VARCHAR(64))

BEGIN

SET @SQL\_STATEMENT = CONCAT('DROP TABLE ', TABLENAME);

PREPARE S1 FROM @SQL\_STATEMENT;

EXECUTE S1;

DEALLOCATE PREPARE S1;

END

Example 36.6:

CREATE PROCEDURE DYNAMIC\_SELECT

(IN SELECT\_STATEMENT VARCHAR(64),

OUT NUMBER\_OF\_ROWS INTEGER)

BEGIN

DECLARE FOUND BOOLEAN DEFAULT TRUE;

DECLARE VAR1,VAR2,VAR3 VARCHAR(100);

DECLARE C\_RESULT CURSOR FOR

SELECT \* FROM SELECT\_TABLE;

DECLARE CONTINUE HANDLER FOR NOT FOUND

SET FOUND = FALSE;

SET @CREATE\_STATEMENT =

CONCAT('CREATE TEMPORARY TABLE SELECT\_TABLE AS (',

SELECT\_STATEMENT, ')');

PREPARE S1 FROM @CREATE\_STATEMENT;

EXECUTE S1;

DEALLOCATE PREPARE S1;

SET NUMBER\_OF\_ROWS = 0;

OPEN C\_RESULT;

FETCH C\_RESULT INTO VAR1, VAR2, VAR3;

WHILE FOUND DO

SET NUMBER\_OF\_ROWS = NUMBER\_OF\_ROWS + 1;

FETCH C\_RESULT INTO VAR1, VAR2, VAR3;

END WHILE;

CLOSE C\_RESULT;

DROP TEMPORARY TABLE SELECT\_TABLE;

END

;

CALL DYNAMIC\_SELECT('SELECT PAYMENTNO, PAYMENT\_DATE, PLAYERNO

FROM PENALTIES', @NUMBER\_OF\_ROWS)

;

SELECT @NUMBER\_OF\_ROWS

Example 37.1:

DELETE

FROM PENALTIES

WHERE PLAYERNO = 44

;

SELECT \*

FROM PENALTIES

;

ROLLBACK WORK

;

COMMIT WORK

Example 37.3:

DELETE FROM PLAYERS WHERE PLAYERNO = 6

;

DELETE FROM PENALTIES WHERE PLAYERNO = 6

;

DELETE FROM MATCHES WHERE PLAYERNO = 6

;

DELETE FROM COMMITTEE\_MEMBERS WHERE PLAYERNO = 6

;

UPDATE TEAMS SET PLAYERNO = 83 WHERE PLAYERNO = 6

;

SHOW GLOBAL VARIABLES LIKE 'COMPLETION\_TYPE'

Example 37.5:

CREATE PROCEDURE NEW\_TEAM ()

BEGIN

INSERT INTO TEAMS VALUES (100,27,'first');

END

;

SET AUTOCOMMIT = 1

;

START TRANSACTION

;

INSERT INTO TEAMS VALUES (200,27,'first')

;

CALL NEW\_TEAM()

;

ROLLBACK WORK

Example 37.6:

UPDATE PENALTIES

SET AMOUNT = AMOUNT + 25

WHERE PAYMENTNO = 4

;

SELECT \*

FROM PENALTIES

WHERE PAYMENTNO = 4

Example 37.7:

SELECT PLAYERNO

FROM PLAYERS

WHERE TOWN = 'Stratford'

;

UPDATE PLAYERS

SET TOWN = 'Eltham'

WHERE PLAYERNO = 7

;

SELECT PLAYERNO, NAME, INITIALS,

STREET, HOUSENO, POSTCODE, TOWN

FROM PLAYERS

WHERE PLAYERNO IN (6, 83, 2, 7, 57, 39, 100)

Example 37.8:

SELECT PLAYERNO

FROM PLAYERS

WHERE TOWN = 'Stratford'

Example 37.9:

UPDATE PENALTIES

SET AMOUNT = AMOUNT + 25

WHERE PAYMENTNO = 4

;

UPDATE PENALTIES

SET AMOUNT = AMOUNT + 30

WHERE PAYMENTNO = 4

Example 37.10:

LOCK TABLE PLAYERS READ

Example 37.11:

DO GET\_LOCK('lock1',0)

;

SELECT GET\_LOCK('lock1',0)

Example 37.12:

SELECT IS\_FREE\_LOCK('lock1')

Example 37.13:

SELECT IS\_USED\_LOCK('lock1')

Example 37.14:

SELECT RELEASE\_LOCK('lock1')