

Exercici 7. Tipus de dades MariaDB/MySQL

L'objectiu d'aquesta activitat és conèixer a fons els tipus de dades més utilitzats en les bases de dades.

Responen a les següents preguntes:

a. Tipus per cadenes de text

1. Quina diferència hi ha entre el tipus de dada CHAR i VARCHAR ?

CHAR s'utilitza si les cadenes sempre són de longitud fixa i si l'espai no es troba en una situació crítica.

VARCHAR s'utilitza si vols optimitzar l'ús d'espai i les cadenes poden tenir longitud variable

2. Quina diferència hi ha entre el tipus de dada CHAR i BINARY ?

CHAR: S'utilitza per emmagatzemar cadenes de caràcters amb una codificació de caràcters específica.

BINARY: S'utilitza per emmagatzemar dades binàries, amb seqüències de bits sense cap codificació de caràcters específica.

3. Què vol dir CHARSET (CHARACTER SET) i COLLATION?

CHARSET(CHARACTER SET): És un conjunt específic de símbols i codificacions de caràcters que es poden utilitzar en una base de dades. Determina quins caràcters es poden emmagatzemar i recuperar en una columna de text o caràcters de la base de dades.

COLLATION: Determina com es comparen i ordenen els caràcters en una operació de cerca o ordenació en una base de dades. Defineix les regles per comparar caràcters té en compte les majúscules/minúscules, accents, etc.

4. Quina diferència hi ha entre el tipus de dades VARCHAR i NVARCHAR ?

Per treballar amb caràcters de diversos idiomes o suport de caràcters especials, és convenient utilitzar NVARCHAR. Si només necessites caràcters del sistema amb VARCHAR és suficient.

Conjunt de caràcters:

VARCHAR: Utilitza conjunt de caràcters del sistema.

NVARCHAR: Utilitza unicode.

Compatibilitat amb idiomes:

NVARCHAR: És més adequat per l'emmagatzematge de caràcters de diversos idiomes ja que representa en caràcter Unicode.

Espai de memòria:

NVARCHAR: Ocupa més espai que VARCHAR per la seva ampla representació de caràcters.

5. Quina és la llargada màxima d'una dada emmagatzemada en un camp CHAR?, i en un VARCHAR?

CHAR: És una dada de longitud fixa, és a dir, sempre ocupa la mateixa quantitat d'espai a la base de dades. Ex: DNI CHAR(9)

VARCHAR: És una dada de longitud variable, si especifiques un camp de VARCHAR(12) i emmagatzema una cadena de 5 caràcters només encadena aquests 5 caràcters sense ocupar la resta. Ex: Cognom VARCHAR(25)

6. Què fa MySQL si s'intenta guardar en un VARCHAR(20) una cadena de 25 posicions?

Només es conservaran els primers 20 caràcters de la cadena i els altres 5 caràcters es descartaran.

7. Què fa MySQL si s'intenta guardar en un CHAR(30) una cadena de 25 posicions ?

La cadena s'omplirà amb caràcters d'espais en blanc per arribar a la longitud especificada. Això es deu a que CHAR té una longitud fixa i sempre ocupa l'espai màxim especificat.

8. Què fa MySQL si s'intenta guardar en un VARCHAR(30) una cadena de 25 ?

La cadena s'enmagatzenarà sense problema i ocuparà l'espai exacte i necessari.

b. Tipus de dades per dates i temps

9. De quins tipus de dades disposa el MySQL per emmagatzemar dates i temps? I Per a quins tipus de valors és adequat cadascun d'ells

DATE: Enmagatzena dates ('2024-01-30').

TIME: Enmagatzema intervals de temps sense la data ('19:30:15').

DATETIME: Enmagatzema dates i hores ('2024-01-30 19:30:15').

TIMESTAMP: Enmagatzema dates i hores amb la capacitat d'actualitzar-se automàticament.

YEAR: S'utilitza per enmagatzemar l'any d'una data en format de 4 dígit ('2024').

10. Quina diferència hi ha entre els tipus DATETIME i TIMESTAMP ?

Es diferencien en el rang de dates, la capacitat de gestionar zones horàries la precisió i la capacitat d'actualitzar-se automàticament amb el temps del sistema.

11. Es poden fer operacions amb dates amb MySQL? Quines ?

CURDATE(): Retorna la data actual sense la hora. `SELECT CURDATE();`

NOW: Retorna la data i l'hora actual. `SELECT NOW();`

DATE_ADD(): Afegeix un interval de temps a una data. `SELECT DATE_ADD(NOW(), INTERVAL 1, DAY);`

DATE_SUB(): Resta un interval de temps a una data. `SELECT DATE_SUB(NOW(), INTERVAL 1, WEEK);`

DATEDIFF(): Calcula la diferència en dies entre dues dates. `SELECT DATEDIFF('2024-02-01', '2024-01-01');`

DATE_FORMAT(): Formata una data com una cadena caràcters segons un format específic. `SELECT DATE_FORMAT (NOW(), '%Y-%m-%d %H:%i:%s');`

YEAR(): Retorna el valor de l'any. `SELECT YEAR(NOW());`

MONTH(): Retorna el valor del mes. `SELECT MONTH (NOW());`

DAY(): Retorna el valor del dia. `SELECT DAY (NOW());`

TIMESTAMPDIFF(): Calcula la diferència entre dues dates o hores en una unitat específica(segons, minuts, hores...) `SELECT TIMESTAMPDIFF (MINUTE, '2024-01-01 13:00:00', NOW());`

c. Tipus de dades numèriques

12. De quins tipus de dades disposa MySQL per a emmagatzemar valors numèrics?

INT(INTEGER): Enmagatzema valors numèrics sencers sense decimals.

TINYINT: Enmagatzema valors sencers petits.

FLOAT: Enmagatzema valors amb decimals de simple precisió.

DOUBLE: Enmagatzema valors amb decimals de doble precisió.

DECIMAL O NUMERIC: Enmagatzema valors amb decimals amb precisió exacte.

13. Quins tipus de nombres es poden guardar amb el tipus DECIMAL o NUMERIC?

Els tipus de nombres que es poden enmagatzemar en tipus DECIMAL o NUMERIC són aquells de precisió fixa. La precisió fixa fa que els calculs mantinguin la precisió desitjada. EX: `DECIMAL(10,2)` Això vol dir que el nombre pot tenir un màxim de 10 dígitos en total, dels quals 2 seràn a la part decimal la part enter tindrà llavors 8 dígitos.

14. Per a què serveixen la precisió i l'escala dels DECIMAL?

La **precisió** es refereix al nombre total de dígitos a la part enter i a la part decimal. Quan declares una dada DECIMAL o NUMERIC especifiques la precisió com a part de la seva definició. La **escala** es refereix al nombre de dígitos de la part decimal d'un nombre, és una subpart de la precisió i especifica la quantitat de dígitos que es

permeten a la part decimal.

La combinació de precisió i escala garanteix que els calculs aritmètics es puguin realitzar amb la precisió desitjada.

15. Què fa MySQL si s'intenta emmagatzemar el valor 12.345 en un camp DECIMAL(5,2) ?

Es produirà un error, ja que hem definit el camp amb una precisió de 5 dígits pel que no pot ser emmagatzemat.

16. Què fa MySQL si s'intenta emmagatzemar el valor 12.34 en un camp NUMERIC(3,2) ?

Es produirà un error, només es permeten 3 dígits en total amb 2 dígits a la part decimal.

17. Quina diferència hi ha entre el tipus FLOAT i el tipus DECIMAL?

DECIMAL: Enmagatzema nombres amb precisió fixa, requereix que especifiquis explícitament la precisió i l'escala quan defineixes un camp. És més adequat per aplicacions on és essencial mantenir una exactitud exacta.

FLOAT: Permet representar nombres amb una àmplia magnitud i precisió, no requereix especificar la precisió i la escala quan defineixes el camp. És més adequat per aplicacions on la magnitud i la velocitat de càlcul són més importants que la precisió exacta.

18. Quina escala fa servir el MySQL sinó s'especifica més que la precisió en un DECIMAL ?

MYSQL assumirà que no hi ha escala especificada o bé que l'escala és 0. Això significa que tots els dígits especificats a la precisió es consideren part de l'enter i no hi ha dígits especificats per la part decimal. EX: DECIMAL(8), MYSQL assumeix que no hi ha part decimal i que tots els 8 dígits són part de l'enter.

19. Com especifiquem que una dada només guardarà valors positius?

Especifiquem amb un modificador **UNSIGNED** juntament amb el tipus de dada numèrica (INT, DECIMAL, FLOAT...). Aquest modificador indica que només s'accepten valors no negatius (zero o valors positius).

20. Què és un ENUM? Posa un exemple de possible ús d'un ENUM.

ENUM és un tipus de dada que pot agafar un conjunt de valors literals especificats durant la definició de la taula. És un tipus de dada que permet limitar els valors que pot prendre una columna a un conjunt concret de possibilitats.