

Indicacions prèvies per als inserts aleatoris en la taula reunions de la BBDD `agenda`

Les taules que com a mínim haurem de fer servir són:

- **contacte**
- **tipusvincle**
- **relacio**
- **reunions**

☐ Cada persona de la taula **contacte** tindrà les següents dades:

Un número de mòbil que serà el seu identificador.

El nom i cognoms de cada persona (en cas que no tingui segon cognom, aquest podrà ser NULL). La seva data de naixement (que podrem calcular de forma aleatòria directament en el full de càlcul on tindrem les columnes de **contacte**) o posteriorment al realitzar el volcatge de dades. També necessitarem un correu electrònic amb els dominis indicats per a cada compte de correu i que haurà de ser únic per a cada **contacte**.

☐ El tipus de vincle entre dos **contactes**, que haurà d'existir prèviament a la taula **tipusvincle** i que inicialment podrà ser:

Comercial (Compra-venda de productes)

Departament (Treballen al mateix departament)

Transport (relació amb enviament de mercaderies)

Assessorament (De formació)

Altres (Relació genèrica o encara no definida)

Aquesta taula disposarà d'una clau primària abstracta o de valor numèric.

☐ La taula **relació**, és el vincle entre dos **contactes** (només podrà existir un tipus de vincle entre dos **contactes**).

☐ La taula **reunions**, on s'enregistraran les diferents reunions dels diferents **contactes** entre els que existeixi un tipus de vincle prèvi, és a dir, si no existeix un registre a la taula **relació** entre dos **contactes**, aquests no podran registrar cap reunió, però en cas afirmatiu podran tenir més d'una reunió en dates diferents. Per això necessitarem la data de la reunió que serà de tipus 'data hora', la durada de la reunió com a valors enters d'hores (amb un màxim de vuit).

El tipus de reunió (Presencial o Telemàtica). L'estat de la reunió, que es representarà per valors entre 1 i 3 (segons sigui Pendent, Realitzada o No realitzada). Un camp `descripció` de tipus GENERATED ALWAYS que canviarà de forma automàtica en funció del valor del camp `estat` de la taula. I per últim crearem un camp de control que anomenarem `data_registre` que automàticament al crear un nou registre haurà d'escriure la data i hora en que s'ha enregistrat.

Atribut de tipus GENERATED column

Els valors de la columna, canvia en funció dels valors d'altres columnes de la mateixa taula.

L'estructura és:

```
GENERATED ALWAYS AS ( condició ) STORED | VIRTUAL;
```

La diferència entre STORED i VIRTUAL, és que en el primer cas s'emmagatzemen les dades i només es calculen la primera vegada o quan canvien les dades de la condició. En el segon cas (virtual) es calculen sempre, en cada lectura o escriptura, per tant és més lent, però ocupa menys espai a la base de dades.

```
DROP TABLE IF EXISTS estalvi;
CREATE TABLE estalvi (
  id INT auto_increment,
  data timestamp DEFAULT current_timestamp,
  ingres DEC(6,2) DEFAULT 0,
  CHECK(( ingres > 0 )),
  despesa DEC(6,2) DEFAULT 0,
  CHECK(( despesa > 0 )),
  acumulat DEC(6,2) GENERATED ALWAYS AS (ingres - despesa) STORED,
  missatge VARCHAR(24) GENERATED ALWAYS AS ( CASE
    WHEN (ingres > despesa) THEN 'Saldo positiu'
    WHEN (ingres < despesa) THEN 'Saldo negatiu'
    ELSE 'Saldo equilibrat'
  ) STORED,
  PRIMARY KEY(id)
);
INSERT into estalvi (ingres, despesa) VALUES (100, 250);
INSERT into estalvi (ingres, despesa) VALUES (100, 50);
INSERT into estalvi (ingres, despesa) VALUES (100, 100);
SELECT * FROM estalvi;
```

SET

La instrucció SET s'utilitza per assignar un valor a una variable de sessió de MySQL.

```
SET @variable = 10;
```

FLOOR()

Aquesta funció s'usa per arrodonir un nombre decimal cap avall fins al nombre enter més proper.

```
SELECT FLOOR(3.8); -- retornarà 3.
```

RAND()

La funció RAND() retorna un nombre decimal aleatori entre 0 i 1.

SELECT RAND(); – pot retornar 0.3456.

- Generar valors aleatoris enters en un rang determinat: Enters entre 5 i 10

SELECT FLOOR(RAND() * (10 - 5 + 1)) + 5 as Valor;

- Generar valors aleatoris entre dues dates:

Data aleatòria entre el 1 de gener i el 31 de desembre de 2022.

SELECT DATE_ADD('2022-01-01', INTERVAL FLOOR(RAND() * DATEDIFF('2022-12-31', '2022-01-01')) DAY);

- Generar valors aleatoris que representin un telèfon de 9 dígit que comenci per 64:

Un número de telèfon aleatori que començarà per '64' i tindrà 7 dígit aleatoris.

SELECT CONCAT('64', LPAD(FLOOR(RAND() * 10000000), 7, '0'));

LPAD(string, length, pad_string):

string: La cadena que vols ampliar amb caràcters a l'esquerra.

length: La longitud total que vols que tingui la cadena després de ser ampliada.

pad_string: (Opcional) El caràcter (o cadena) que s'utilitzarà per omplir a l'esquerra la cadena string.

Per defecte, és un espai en blanc.

En la nostra sentència anterior:

FLOOR(RAND() * 10000000)

Aquesta part genera un nombre aleatori entre 0 i 9999999 (excloent 10000000).

LPAD(FLOOR(RAND() * 10000000), 7, '0')

Aquesta part amplia el nombre generat a la part anterior amb zeros a l'esquerra fins que tingui una longitud de 7 dígit.

CONCAT('64', LPAD(...))

Finalment, s'uneix '64' amb el resultat de la funció LPAD(), generant així un número de telèfon de 9 dígit que comença amb '64' i té 7 dígit aleatoris a la resta de la cadena.

Per exemple, si el nombre generat és '123456', la funció LPAD() el convertirà en '0123456' perquè tingui una longitud total de 7 dígit, i després s'unirà amb '64' per formar '640123456'. Així es garanteix que el número tingui 9 dígit i comenci amb '64'.

DATE_FORMAT()

Aquesta funció s'utilitza per formatejar una data i hora en una cadena de text amb un format específic.

```
SELECT DATE_FORMAT( NOW(), '%Y-%m-%d');
```

Retorna la data actual en el format 'YYYY-MM-DD'

NOW()

La funció NOW() retorna la data i hora actual del sistema.

```
SELECT NOW(); – pot retornar '2024-02-20 14:30:00'.
```

INTERVAL

L'expressió INTERVAL s'utilitza per afegir o restar una quantitat específica de temps a una data o hora.

```
SELECT NOW() + INTERVAL 1 DAY; – retorna la data de demà.
```

CASE

L'expressió CASE s'utilitza per avaluar condicions i retornar un valor específic basat en el resultat de les condicions.

```
CREATE TABLE estudiants (  
  id INT auto_increment PRIMARY KEY,  
  nota INT,  
  CHECK (( nota >= 0 AND nota <= 10))  
);
```

```
INSERT IGNORE INTO estudiants (nota) VALUES  
(-3), (5), (8), (9), (12);
```

```
SELECT * FROM estudiants;
```

```
+-----+-----+
| id | nota |
+-----+-----+
| 1 | 5 |
| 2 | 8 |
| 3 | 9 |
+-----+-----+
3 rows in set (0,00 sec)
```

```
SELECT id,
       CASE
         WHEN nota >= 9 THEN 'Excel·lent'
         WHEN nota >= 7 THEN 'Notable'
         ELSE 'Aprovat'
       END AS qualificacio
FROM estudiants;
```

```
+-----+-----+
| id | qualificacio |
+-----+-----+
| 1 | Aprovat |
| 2 | Notable |
| 3 | Excel·lent |
+-----+-----+
3 rows in set (0,00 sec)
```