√ 1 row inserted.

Inserted row id: 10 (Query took 0.0013 seconds.)

✓ 1 row inserted.

Inserted row id: 10 (Query took 0.0013 seconds.)

✓ 1 row inserted.

Inserted row id: 10 (Query took 0.0013 seconds.)

✓ 1 row inserted.

Inserted row id: 10 (Query took 0.0013 seconds.)

✓ 1 row inserted.

Inserted row id: 10 (Query took 0.0013 seconds.)

✓ 1 row inserted row id: 10 (Query took 0.0013 seconds.)

✓ 1 row inserted row id: 10 (Query took 0.0013 seconds.)

✓ 1 row inserted row id: 10 (Query took 0.0013 seconds.)

✓ 1 row inserted row id: 10 (Query took 0.0013 seconds.)

✓ 1 row inserted row id: 10 (Query took 0.0013 seconds.)

✓ 1 row inserted row id: 10 (Query took 0.0013 seconds.)

✓ 1 row inserted row id: 10 (Query took 0.0013 seconds.)

✓ 1 row inserted row id: 10 (Query took 0.0013 seconds.)

✓ 1 row inserted row id: 10 (Query took 0.0013 seconds.)

✓ 1 row inserted row id: 10 (Query took 0.0013 seconds.)

✓ 2 row inserted row id: 10 (Query took 0.0013 seconds.)

✓ 2 row inserted row id: 10 (Query took 0.0013 seconds.)

✓ 3 row inserted row id: 10 (Query took 0.0013 seconds.)

✓ 3 row inserted row inserted row id: 10 (Query took 0.0013 seconds.)

✓ 3 row inserted row inser

INSERT INTO users (nom, prenom, email, mot_de_passe, adresse, telephone, specialite, classe, statut_actuel) VALUES ('Dupont', 'Jean', 'jean.dupont@mail.com', 'df6b9fb15cfdbb7527be5a8a6e39f39e572c8ddb943fbc79a943438e9d3d85ebfc2ccf9e0', 'Tunis', '22123456', 'Informatique', 'Master', 'Étudiant');









WhySQL returned an empty result set (i.e. zero rows). (Query took 0.0027 seconds.)

CREATE UNIQUE INDEX idx_users_email ON users(email);

[Edit inline][Edit][Create PHP code]

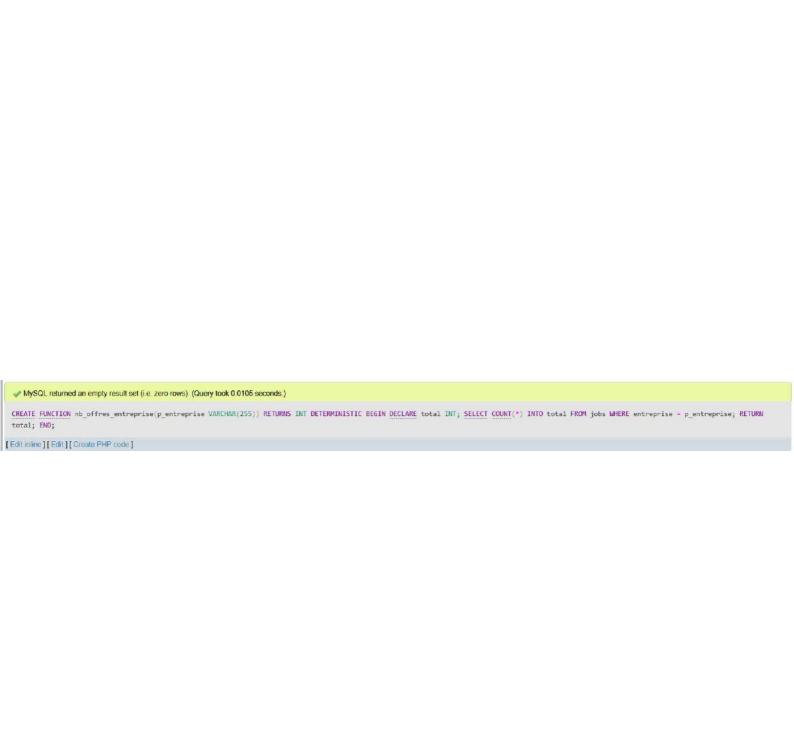
Note: #1831 Duplicate index 'idx_users_email'. This is deprecated and will be disallowed in a future release

WhySQL returned an empty result set (i.e. zero rows). (Query took 0.0008 seconds.)

CREATE INDEX idx_jobs_titre ON jobs(titre);

[Edit inline][Edit][Create PHP code]









CREATE PROCEDURE sp_list_jobs() BEGIN -- Déclarations : obligatoirement directement après BEGIN DECLARE done INT DEFAULT 0; DECLARE v_titre VARCHAR(255); DECLARE v_entreprise VARCHAR(255); DECLARE v_lieu VARCHAR(255); -- Déclaration du curseur DECLARE cur_jobs CURSOR FOR SELECT titre, entreprise, lieu FROM jobs; -- Handler pour fin de curseur DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1; -- Maintenant seulement les autres instructions (temp table, OPEN, LOOP...) DROP TEMPORARY TABLE IF EXISTS tmp_jobs; CREATE TEMPORARY TABLE tmp_jobs (offre_info VARCHAR(500)) ENGINE=MEMORY; OPEN cur_jobs; read_loop: LOOP FETCH cur_jobs INTO v_titre, v_entreprise, v_lieu; If done = 1 THEN LEAVE read_loop; END IF; INSERT INTO tmp_jobs (offre_info) VALUES (CONCAT('Poste: ', IFNULL(v_titre,''), ' | Entreprise: ', IFNULL(v_nentreprise,''), ' | Lieu: ', IFNULL(v_lieu,''))); END LOOP read_loop; [...]

[Edit]





