

# PROJET BASES DE DONNEES - REQUÊTES

Chang GAO et Tom SUBLET

Gr TD 1.2 - DUT Info

IUT Metz 2019-2020

## Info

Ce fichier est écrit en **Markdown Extended**.

## 1. Bonne(s) réponse(s)

Requête :

```
SELECT rep_proposee.lib_reponse AS "Réponse Q03"  
FROM rep_proposee  
NATURAL JOIN question  
WHERE rep_proposee.etat_rep = 1  
AND question.no_question = "Q03";
```

Résultat :

Réponse Q03
Nino Ferrer

## 2. Réponse donnée par STAUB Guillaume à la question 4 lors de la session 4

---

Requête :

```
SELECT lib_reponse AS "Réponse par STAIB Guillaume", etat_rep AS "Juste ?"  
FROM rep_donnee  
NATURAL JOIN rep_proposee  
NATURAL JOIN session  
NATURAL JOIN personne  
WHERE rep_donnee.no_question = "Q04"  
      AND nom_pers = "STAUB"  
      AND prenom_pers = "Guillaume"
```

Résultat :

Réponse par STAIB Guillaume	Juste ?
6	0

### 3. Pourcentage des utilisateurs qui ont tenté plusieurs fois le même questionnaire

Requête :

```
CREATE TEMPORARY TABLE multiplesTry
SELECT COUNT(DISTINCT s1.no_pers) AS "nbIndivWhoTries"
FROM session AS s1
INNER JOIN session AS s2 ON s1.no_session != s2.no_session
                        AND s1.no_quest = s2.no_quest
                        AND s1.no_pers = s2.no_pers
;
SELECT nbIndivWhoTries/COUNT(no_pers) AS "% des utilisateurs"
FROM multiplesTry, session
;
DROP TABLE multiplesTry
```

Résultat :

% des utilisateurs
0.1429



#### Note

C'est à l'affichage que l'on affichera le pourcentage en %, donc on ne multiplie pas par 100 lors du calcul.

## 4. Les personnes qui ont répondu en 2019 à des questionnaires sur le sport

---

Requête :

```
SELECT DISTINCT nom_pers AS "Nom", prenom_pers AS "Prénom"  
FROM session  
NATURAL JOIN questionnaire  
NATURAL JOIN personne  
NATURAL JOIN theme  
WHERE libelle_theme = "Sport"  
AND YEAR(date_session) = 2019  
ORDER BY date_session DESC, nom_pers, prenom_pers;
```

Résultat :

Nom	Prénom
ZANZANA	Nathan
MARTEAU	Catherine
STAUB	Guillaume

## 5. Le pourcentage de réponses correctes à la question 2

Requête :

```
CREATE TEMPORARY TABLE correctAnswers
SELECT COUNT(no_ordre) AS "nbCorrect"
FROM rep_donnee
NATURAL JOIN rep_proposee
WHERE no_question = "Q02"
      AND etat_rep = 1
;
CREATE TEMPORARY TABLE totalAnswers
SELECT COUNT(no_ordre) AS "nbAnswers"
FROM rep_donnee
WHERE no_question = "Q02"
;
SELECT nbCorrect/nbAnswers
      AS "% de bonne réponses"
FROM correctAnswers, totalAnswers
;
DROP TABLE correctAnswers
;
DROP TABLE totalAnswers
```

Résultat :

% de bonne réponses
1.0000



### Note

C'est à l'affichage que l'on affichera le pourcentage en %, donc on ne multiplie pas par 100 lors du calcul.

## 6. Le pourcentage de non-réponses à la question 2

Requête :

```
CREATE TEMPORARY TABLE noAnswers
SELECT COUNT(no_session) as nbNoAnswers
FROM se_compose
NATURAL JOIN session
NATURAL LEFT JOIN rep_donnee
WHERE no_question = "Q02"
      AND rep_donnee.no_ordre IS null
;
CREATE TEMPORARY TABLE totAnswers
SELECT COUNT(no_session) as nbAnswers
FROM se_compose
NATURAL JOIN session
WHERE no_question = "Q02"
;
SELECT nbNoAnswers/nbAnswers AS "% de non-réponses"
FROM noAnswers, totAnswers
;
DROP TABLE noAnswers
;
DROP TABLE totAnswers
```

Résultat :

% de non-réponses
0.0000



### Note

Si une réponse n'est pas donnée par un candidat, cette dernière ne possède pas de `no_session`, de `no_question` ni de `no_ordre` dans la table `rep_donnee`. D'où le `rep_donnee.no_ordre IS null`.

## 7. Les questions auxquelles les utilisateurs n'ont jamais répondu lors de leurs différentes sessions

---

Requête :

```
SELECT no_question AS "Numéro", libelle AS "Question"
FROM question
WHERE no_question NOT IN(
    SELECT no_question
    FROM rep_donnee
)
```

Résultat :

Numéro	Question
Q07	La préhistoire débute :

## 8. Le pourcentage des questionnaires correspondant aux sessions de septembre 2019 portant sur le sport ou la bière

Requête :

```
CREATE TEMPORARY TABLE sessionFilter
SELECT COUNT(no_quest) AS sessionFiltred
FROM questionnaire
NATURAL JOIN theme
NATURAL JOIN session
WHERE YEAR(date_session)="2019"
      AND MONTH(date_session)="09"
      AND (libelle_theme = "Sport" OR libelle_theme = "Biere")
;
CREATE TEMPORARY TABLE sessionAll
SELECT COUNT(no_quest) AS sessionNb
FROM questionnaire
NATURAL JOIN session
WHERE YEAR(date_session)="2019"
      AND MONTH(date_session)="09"
;
SELECT sessionFiltred/sessionNb AS "% des questionnaires"
FROM sessionFilter, sessionAll
;
DROP TABLE sessionFilter
;
DROP TABLE sessionAll
```

Résultat :

% des questionnaires
1.0000



### Note

C'est à l'affichage que l'on affichera le pourcentage en %, donc on ne multiplie pas par 100 lors du calcul.



## 9. Les questions (classées par numéro) auxquelles n'a pas répondu DUVAL Maxime lors de sa session 3

Requête :

```
SELECT DISTINCT no_question AS "Numéro", libelle AS "Question"
FROM se_compose
NATURAL JOIN session
NATURAL JOIN personne
NATURAL JOIN question
LEFT JOIN rep_donnee USING (no_question)
WHERE rep_donnee.no_session IS null
      AND nom_pers = "DUVAL"
      AND prenom_pers = "Maxime"
      AND session.no_session = "S03"
ORDER BY no_question
```

Résultat :

Numéro	Question
Q07	La préhistoire débute :



### Note

Si une réponse n'est pas donnée par un candidat, cette dernière ne possède pas de no\_session, de no\_question ni de no\_ordre dans la table `rep_donnee`. D'où le `rep_donnee.no_session IS null`.

## 10. Pourcentage des questions ont une réponse juste proposée en 1e position

Requête :

```
CREATE TEMPORARY TABLE corQuestions
SELECT COUNT(DISTINCT no_question) as nbCorQuestions
FROM rep_proposee
WHERE no_ordre = 1
      AND etat_rep = 1
      --un BOOL étant traduit en TINYINT(1), 1 équivaut à la valeur vraie
;
CREATE TEMPORARY TABLE totQuestions
SELECT COUNT(no_question) as nbQuestions
FROM question
;
SELECT nbCorQuestions/nbQuestions AS "% de questions"
FROM corQuestions, totQuestions
```

Résultat :

% des questions
0.5714



### Note

C'est à l'affichage que l'on affichera le pourcentage en %, donc on ne multiplie pas par 100 lors du calcul.

## 11- Vérification de la cohérence : Réponses enregistrées pour un utilisateur ne correspondant pas à une des propositions

Requête :

```
SELECT *  
FROM rep_donnee  
WHERE no_ordre NOT IN (  
    SELECT no_ordre  
    FROM rep_proposee  
)
```

Résultat :

no_session	no_question	no_ordre



### Note

Notre jeu d'essai étant cohérent, la requête ne renvoie aucune valeur.

## 12- Vérification de la cohérence : Questions ne comportant aucune bonne réponse

Requête :

```
SELECT no_question, libelle
FROM rep_proposee
NATURAL JOIN question
GROUP BY no_question
HAVING MAX(etat_rep) != 1
--un BOOL étant traduit en TINYINT(1), 1 équivaut à la valeur vraie
```

Résultat :

no_question	libelle



### Note

Notre jeu d'essai étant cohérent, la requête ne renvoie aucune valeur.