

06/7/9/11/12/14/15/20/41/44

Collège de Candolle

ÉPREUVE REGROUPEE

Discipline: INFORMATIQUE

Date: 13.06.2024

Cours: 2 IN DP 02 / 08

Durée: 105 minutes

Maîtres: JC BARROS

Matériel autorisé: aucun (aide-mémoire inclus à la fin de l'épreuve)

20 minutes

Nom et prénom:

Total: 66 points

Bases de données (29 pts)

Ex. 1. Questions sur une base de données SQL (5 pts)

a. Quelle instruction SQL est utilisée pour renvoyer uniquement des valeurs différentes ? (1 pt)

DISTINCT

b. Quel(s) mot(s)-clé SQL doit-on utiliser pour trier le résultat d'une requête ? (1 pt)

ORDER BY

c. Les colonnes des tables sont également connues sous le nom de (1 pt):

champ

d. A quoi sert une clé étrangère ? (2 pt)

à faire un lien avec une autre table
(contenant une def primaire de même valeur)

Ex. 2. (6 pts)

Que signifient les sigles SQL et NoSQL ? (1 pts)

Structured Query Language

Not Only SQL

Remplissez le tableau ci-dessous avec des propriétés des Bases de Données SQL et NoSQL

	SQL	NoSQL
très structuré et normalisé ? (oui / non) (1 pts)	oui	non
toujours stocké sous forme de tables ? (oui / non) (1 pts)	oui	non
Adapté au Big Data (oui / non) (1 pts)	non	oui
Exemples d'application (écrivez les numéros des exemples ci-dessous) (2 pts)	1 / 2	3 / 4

Exemples d'application à insérer dans la dernière ligne du tableau ci-dessus:

1. gestion des comptes dans une banque
2. gestion des notes dans une école
3. détermination des meilleurs résultats pour une recherche sur le web (exemple Google)
4. analyse des liens sociaux entre les utilisateurs de différents réseaux sociaux

Ex. 3. Schéma entité-relation vers schéma relationnel (8 pts)

À partir du schéma d'entité-relation ci-dessous, élaborer le schéma du modèle relationnel. Indiquez-y clairement toutes les clés primaires (PK) et étrangères (FK).

Schéma E-R:

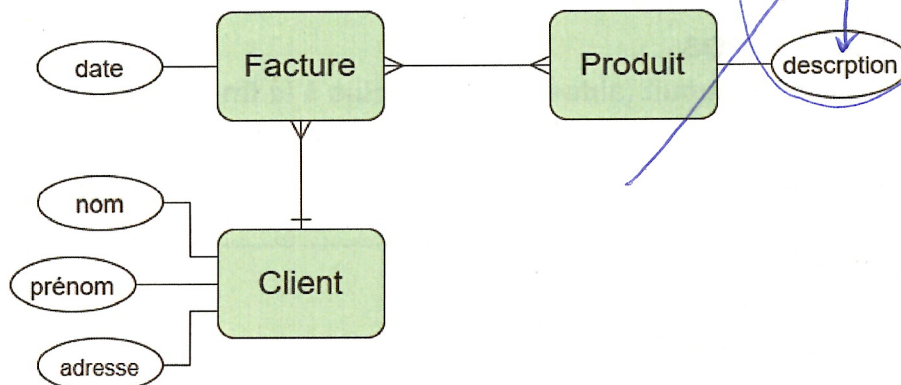
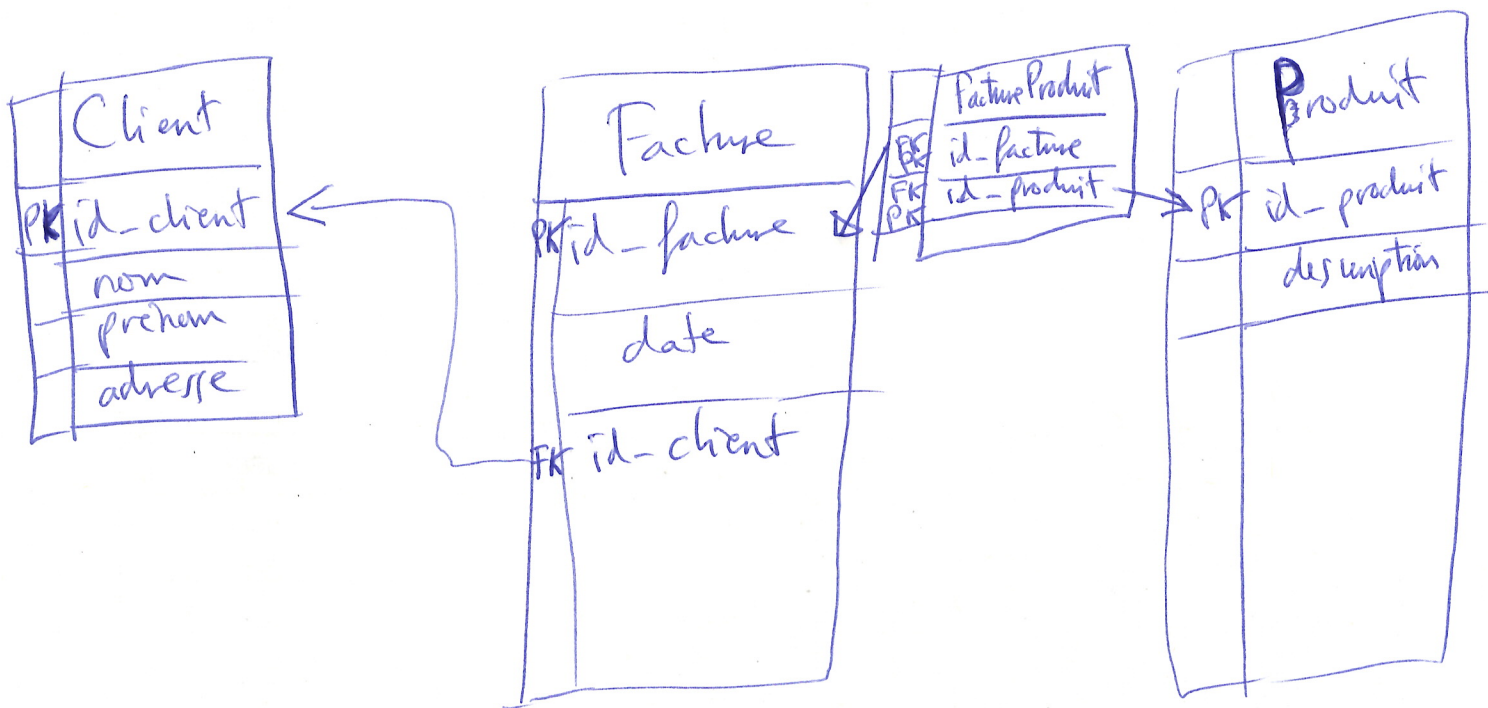


Schéma relationnel (tables):



Ex. 4. Dans une base de données correspondant au schéma ci-dessus (Ex. 2), explique de manière synthétique ce qu'affichent les requêtes suivantes ? (4 pts)

SELECT * FROM Client WHERE id_client=4;

affiche tous les champs de la table Client pour le client dont id_client vaut 4.

SELECT DISTINCT date FROM Facture;

affiche chaque date présente dans la table Facture une seule fois (sans répétition)

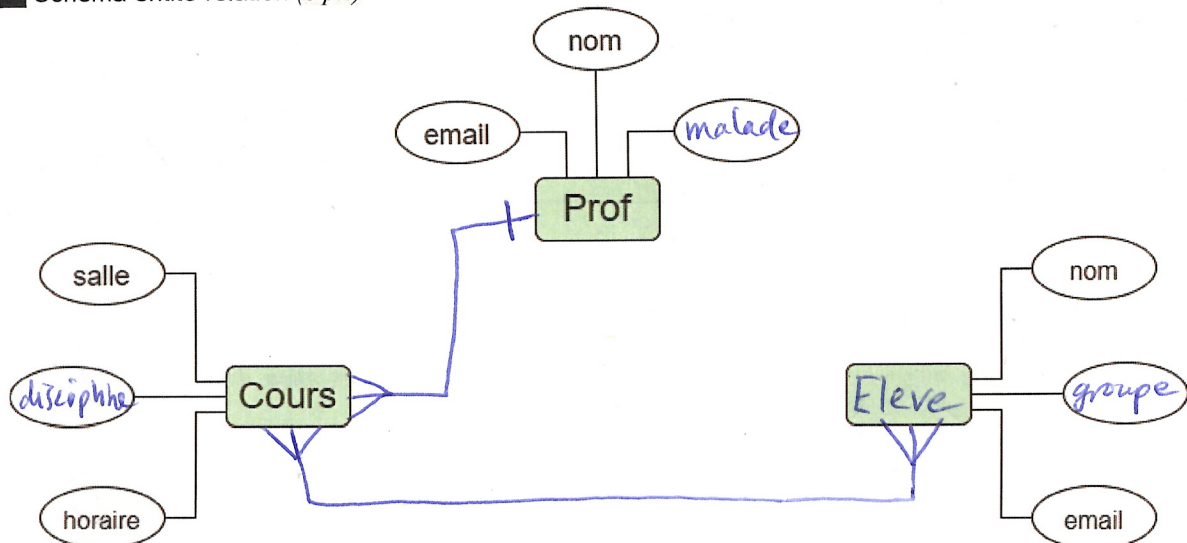
SELECT prenom, nom FROM Client WHERE nom > 'H' AND nom < 'I';

affiche les champs prenom et nom de la table Client dont le nom commence par 'H'

SELECT id_facture, date FROM Facture ORDER BY date DESC LIMIT 10;

affiche les champs id_facture et date de la table Facture mais seulement les 10 ~~plus récents~~ plus récents.

Ex. 5. Schéma entité-relation (6 pts)



Situation :

Un directeur d'école souhaite gérer chaque cours en fonction du prof qui le donne et des élèves qui le suivent.

- Ecrivez le nom de la 3^{ème} entité sur le schéma ci-dessus. (0,5 pt)
- Ajoutez les 3 attributs suivants dans le schéma (là où c'est le plus indiqué): (1,5 pts)
 - discipline
 - malade
 - groupe (administratif)
- Ajoutez les relations utiles entre les différentes entités en représentant les cardinalités de chaque relations comme dans l'EX. 3. (2 pts)
- Justifiez chaque cardinalité des relations tracées au point c. (2 pts)

entités en relation (n, 1)

justification de la cardinalité

Cours / Prof (n, 1) → 1 prof donne plusieurs cours mais 1 cours est donné par 1 seul prof

Cours / Eleve (n, n) → 1 cours contient plusieurs élèves 1 élève suit plusieurs cours

Sécurité (27 pts)

Ex. 6. Décrire brièvement une technique pour assurer chacun des principes de suivants: (4 pts)

- CONFIDENTIALITÉ: *chiffrement*
- INTÉGRITÉ: *bit de parité, check sum, hash*
- AUTHENTICITÉ: *TLS, signature, service d'authentification*
- DISPONIBILITÉ: *alimentation redondante, antivirus, serveurs redondants*

Ex. 7. Chiffrez le mot "SALUT" avec les deux chiffrements symétriques / clés indiqués. (2 pts)

	version1	version 2
méthode de chiffrement	Vigenère	César
texte clair	SALUT	SALUT
clé	AB	C
texte chiffré	<i>SBLVT</i>	<i>UCNWW</i>

Ex. 8. Ecrivez ou entourez les mots adéquats. (6 pts)

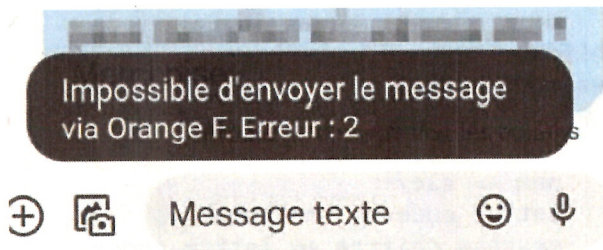
Le chiffrement asymétrique utilise *2* clés. Si Bob veut envoyer un message à Alice, il va *chiffrer* son message avec la clé *publique* / *privée* de *Bob* / *Alice*.

Quand Alice reçoit le message de Bob, pour le lire, elle utilise la clé *publique* / *privée* de *Bob* / *Alice*.

Ex. 9. Cochez une case par ligne dans la colonne correspondante. (5 pts)

	Descriptions	Chiffrement symétrique	Chiffrement asymétrique
1	plus sûr	☒	☐
2	plus performant (rapide) à l'exécution	☒	☐
3	secret partagé nécessaire <u>avant</u> l'utilisation.	☒	☐
4	plus gourmand en ressources (temps / mémoire) à l'exécution	☐	☒
5	capacité à créer un secret partagé en ligne	☐	☒

Ex. 10. Écrivez à côté de chaque image le principe de sécurité (principal) concerné (5 pts):

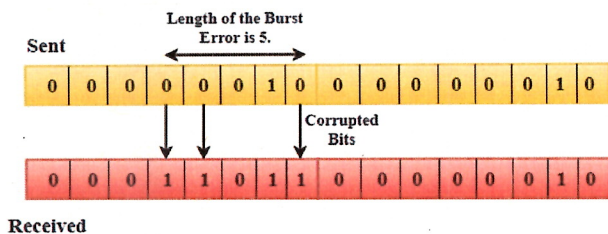


disponibilité

Certificat

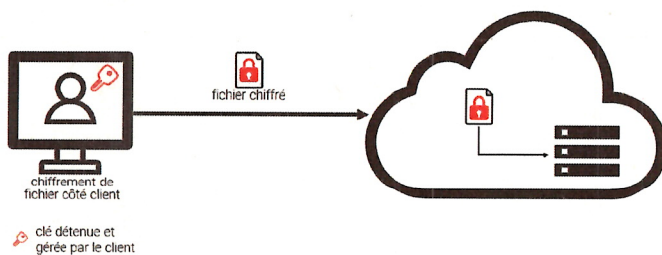
eduge.ch		R3	ISRG Root X1
Nom du sujet			
Nom courant	eduge.ch		
Nom de l'émetteur			
Pays	US		
Organisation	Let's Encrypt		
Nom courant	R3		
Validité			
De	Mon 05 Mar 2024 16:06:12 GMT		

authentification

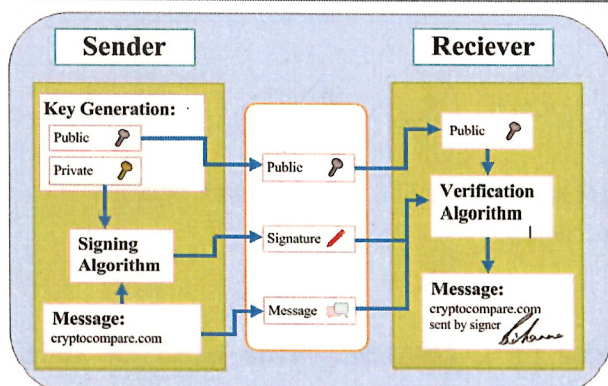


intégrité

Received



confidentialité



authentification / non répudiation

Ex. 11. (5 pts)

a. A quelle méthode de chiffrement correspond le code python suivant ? (2 pts)

CESAR

```
1 def chiffrer(message, clef):
2     message_chiffre = ""
3     for x in message:
4         num = ord(x)
5         num += clef
6         lettre_codee = chr(num)
7         message_chiffre += lettre_codee
8
9     return message_chiffre
10
11 print(chiffrer("message", 3))
```

```
>>> %Run -c $EDITOR_CONTENT
phvvdjh
```

b. Saisissez ci-dessous la(les)

modification(s) du programme pour qu'il affiche **qiwweki** (3 pts)

11 → print(chiffrer("message", 4))

Récurtivité (10 pts)

Ex. 12. Deux parties de code manquent à ce programme (#proposition1 et #proposition2) (10 pts)

```
1 def puissance(x, n):
2     """
3     calcule x à la puissance n
4     """
5     if n == 0:
6         return # proposition 1
7     else:
8         return # proposition 2
9
10
11 print(puissance(10, 6))
```

Console

```
>>> %Run recursif_puissance_squelette.py
```

1000000

a. Que faut-il écrire au lieu de # proposition 1 (ligne 6) et # proposition 2 (ligne 8) pour que ce programme récurif donne la réponse affichée dans la console? (3 et 4 pts)

i. # proposition 1 → 1

ii. # proposition 2 → x * puissance(x, n-1)

b. Est-ce que cette proposition remplaçant entièrement les lignes 5 et 6 fonctionnerait également? Si oui, justifier; sinon, modifier pour que ça marche... (3 pts)

```
5 if n == 1:
6     return n
```

FIN DE L'ÉPREUVE (aide-mémoire suit)