1) Citer SGBD, réputé pour durée support.
PostgreSQL

2) Commande netstat -lnt , comment on s'en sert en pratique et pk.
La commande netstat -lnt | grep :$PGSQL_PORT permet de retourner des informations sur les états du réseau. Ici, on vérifie plus particulièrement que le port spécifié est libre, et s'il ne l'est pas, un message devrait être retourné.

3) Vérifier qu'un serveur de BD s'est lancé correctement.
En utilisant la commande ls /tmp/pgsql/data, 2 nouveaux fichiers viennent sont crées : postmaster.opts et postmaster.pid (fichiers qui n'existaient pas avant de lancer le serveur). Aussi, avec la commande netstat -lnt | grep : $PGSQL_PORT, 2 lignes d'informations sont affichées, ce qui n'était pas le cas quand le serveur n'était pas encore lancé.

4) CREATE TRIGGER D
BEFORE INSERT ON T1
FOR EACH ROW
WHEN NEW A IN (SELECT A FROM T)
BEGIN
SELECT raise(ignore); --Lève une erreur
END;
Proposer contrainte d'intégrité équivalente et nommer type de cette contrainte.

Alter Table T
ADD UNIQUE (A);
On évite les doublons

5) Reformuler contrainte de saisie non-nulle avec déclencheur.
CREATE OR REPLACE TRIGGER non-nulle
BEFORE INSERT ON T
BEGIN
IF A IS NOT NULL THEN
INSERT INTO T VALUES (A)
END IF;
END;
$$ LANGUAGE plpgsql ;

6) Fonction qui convertit des euros en dollars (avec un taux de 1,35).
CREATE OR REPLACE FUNCTION conversion ()
RETURNS TRIGGER AS $$
BEGIN
UPDATE Notation SET somme = somme *1.35 ;
RETURN NEW ;
end if;
END ;
$$ LANGUAGE plpgsql;

CREATE TRIGGER eurosDollar
AFTER INSERT ON Salaire
EXECUTE PROCEDURE conversion();

7) Les deux raisons motivent l'utilisation transactions?
Transaction->éxécutées de façon isolée
Transaction->en cas de panne MAJ ignorées

8) "cinq neuf" ?
-> Les "cinq neuf" correspondent à une disponibilité de 99,999%. Ce pourcentage correspond au temps où un service est assuré sur une période donnée. Dans le cas d'une année, un service "cinq neuf" n'est considérée indisponible que 5,26 minutes.

9) Synthèse de l'ensemble du cours en décrivant les concepts abordés et la façon dont ils s'articulent.
-> On a aussi bien vu des éléments de sécurité (view, droits) que des éléments qui permettaient un certain contrôle des données (view, trigger). Ces 2 aspects permettent de mieux aborder le métier d'administrateur de bases de données.
On a aussi abordé la performance avec les Index et les transactions afin de permettre un meilleur accès aux données stockées.

10) Acronymes SGBD et DBMS?
-> SGBD : Systeme de Gestion de Base de Données
DBSM : Data Base System Management

11) En qui consiste métier d'administrateur de BD ? différents volets d'intervention ? aspects principaux ?
->En quoi ça consiste :
Personne permettant le fonctionnement et l'évolution d'un système d'information utilisant un SGBD.
2 catégories : (orientation contenu : modélisation, procédures, scripts// orientation contenant : contraintes, davantage d'administration)
Nécessite des compétences en : développement (langage SQL), modélisation (connaissance fonctionnelle des données dans la base) et système et réseaux (interactions SGBD/OS, protocoles réseaux)

Volets d'intervention :
- >Intégrité (vérifie cohérence des données, développe contraintes et procédures, effectue la validation)
Disponibilité (plan de sauvegarde et de récup en cas de panne, cohérence des sauvegardes, archivage données, contact du support technique)
Performance (Optimisation de la BD, ajustement paramètres SGBD, Elaboration de statistiques etc)
Sécurité (Gestion des privilèges, coordonne mesures de sécurité, surveille accès à la BD)

Aspects principaux :
Tâches de mises en service : Validation schéma de données, installation, configuration et administration du SGBD, installation ressources matérielles, uniformisation de l'usage de la BD
Activités de suivie(Maintenance : résolution problèmes, mise à jour, archivage //surveillance : rapports de performance, vérifie état du matériel, accès aux données // Tâches préventives : Etude d'autres environnement pour migrer, audit, prévision besoin futurs...)

12) Syntaxe qui permet de réaliser une requête DDL(création de table par exemple). A quel(s) élément(s) faut-il faire attention lors de la recherche de cette information? Pourquoi?
-> Il faut faire attention au SGBD que l'on utilise puisque le type de requête diffère.

Par exemple, sur PostgreSQL, on utilise une fonction spécifique, sur Oracle Database, il y a des instructions DML mais pas DDL, tandis que sur mySQL, il y a une instruction DML.

(c'est ce que j'ai compris avec le cours)

-> Sur google :

Les déclencheurs DDL et les déclencheurs DML sont utilisés dans des buts différents.
Les déclencheurs DML sont associés aux instructions INSERT, UPDATE et DELETE et contribuent à l'application de règles métier et à l'intégrité des données lorsque le contenu de tables ou de vues est modifié.
Les déclencheurs DDL fonctionnent sur les instructions CREATE, ALTER, DROP et autres instructions DDL et procédures stockées qui effectuent des opérations de type DDL. Ils sont utilisés pour exécuter des tâches administratives et appliquer des règles métier qui concernent les bases de données. Ils s'appliquent à toutes les commandes d'un même type dans une base de données, ou sur un serveur.

13) Expliquer le problème qui se pose dans la requête suivante si l'attribut A possède des valeurs nulles mais pas l'attribut B.
SELECT * FROM T WHERE A NOT IN (SELECT B FROM U)

-> Si A possède des attributs null mais pas B, alors il y aura certains résultats qui ne seront pas retournés. (???)
En gros on pourra afficher quand tous les A seront null + les cas où B n'a pas les mêmes valeurs.

14) Comparer les deux SGBD vus en TP (force et faiblesses respectives par exemple).
PostgreSQL: gros volume de données, plus avancé que mySQL, volonté de supporter le standard, optimisateur de requêtes efficace, réputé pour être lent, en particulier pour les requêtes simples, SGBD libre le plus avancé du monde, -9% d'offres d'emploi du 13/2/2013 au 13/02/2015
MySQL : petit volume de données, SGBD libre le plus populaire du monde, +21% d'offres d'emploi du 13/2/2013 au 13/02/2015
-> Performance actuelle de MySQL et PostgreSQL comprabales

15) 4 contraintes de types distincts ; indiquer le niveau auquel elle agit,son effet et donner le mot-clé SQL associé.
Contraintes sur les attributs et les tuples :
Saisie non-nulle (NOT NULL).
Unicité (UNIQUE).
Intégrité de domaine (CHECK).
Contraintes sur les clés (niveau table) :
Intégrité d'entité (PRIMARY KEY).
Intégrité référentielle (FOREIGN KEY).
Assertions générales (niveau schéma).

16) Expliquer et discuter l'intérêt des vues ?
-> Dissocier l'accès aux données de l'organisation conceptuelle (favorise l'indépendance des données et des programmes).
-> Masquer certaines informations et restreindre des droits d'accès.
-> Faciliter l'accès aux données (requêtes plus naturelles).
-> Améliorer les performances des requêtes.

17) Avantages et inconvénients d'une vue matérialisée par rapport à vue simple?
Avantages :
restreindre accès , facilite accès aux données, mises à jour automatiques ->augmentation des performances des requêtes, synchronisation entre les tables de base et la vue matérialisée

Vue V = RequêteVue(T1, T2, ..., Tn).
Les attributs de V sont ceux du résultat de la requête.
Exécute RequêteVue et met le résultat dans V.
Les requêtes utilisent V comme un table
Défauts : Mises à jours automatiques qui provoque temps de calculs longs + synchronisation entre les tables et la vue matérialisée.
(La synchronisation peut être à la fois bien et mauvaise, si beaucoup de calcul => ralenti mais si peu -> Maj automatique)

Inconvénients :
La vue V peut être très large.
Les modifications de T1, T2, ..., Tn nécessite de recalculer ou mettre à jour V..
- Des modifications peuvent invalider la vue matérialisée

Le gain en performance dépend de plusieurs facteurs :
Taille des données.
Complexité de la vue.
Nombre de requêtes utilisant la vue.
Nombre de modifications affectant la vue.
Maintenance incrémentale vs. calcul complet.
Lié au compromis lecture/écriture.

18) Définir ce qu'est une transaction.
Une transaction est une séquence d'une ou plusieurs requêtes SQL considérées comme un élément unitaire.

19) ACID
Atomicité : chaque transaction est soit prise en compte entièrement ou complètement ignorée en cas d'échec (principe du tout-ou-rien).
Cohérence : La sérialisation assure que les contraintes sont toujours respectées
Isolation : Exécution seule (aucunes interactions). Aucune dépendance possible entre les transactions. ->exécution simultanée de transactions produit le même état que celui qui serait obtenu par l'exécution en série des transactions
Durabilité : si le système subit une panne, tous les effets de la transaction concernée restent dans la bases de données.

20) Expliquer les différents types de lectures posant problème (une phrase prècise et synthétique par lecture)

Lecture sale : Une transaction lit des données écrites par une transaction concurrente sans commit.
Lecture non-reproductible : Une transaction relit des données qu'elle a lu précédemment et trouve que les données ont été modifiées par une autre transaction (validée depuis la lecture initiale).
Lecture fantôme : Lire une valeur qui n'est plus la même.

21) Compléter tableau en indiquant les garanties offertes par chaque niveau d'isolation (oui si le problème peut apparaître, non sinon).

READ UNCOMMITED-> lecture sale + non reproductible + lecture fantome
READ COMMITED -> non reproductible(un élément déjà lu ne change pas de valeur) + lecture fantome
REPEATABLE READ -> lecture fantome (pas de garantie de sérialisation)
SERIALIZABLE -> Protège de tout (LECTURE SALE ->une donnée **sale** est une donnée qui a été écrite par une transaction non-validée (sans commit))

22) Supposons que vous avez oublié comment on commence une transaction en SQL et que vous décidez d'utiliser un moteur de recherche pour trouver cette information.Que doit-elle inclure ?

START TRANSACTION
[transaction_characteristic [, transaction_characteristic] ...]

transaction_characteristic:
WITH CONSISTENT SNAPSHOT
READ WRITE
READ ONLY
BEGIN [WORK]
COMMIT [WORK] [AND [NO] CHAIN] [[NO] RELEASE]
ROLLBACK [WORK] [AND [NO] CHAIN] [[NO] RELEASE]
SET autocommit = {0 | 1}

PARTIE COURS

CONTRAINTES(statique)

Contraint les états de la base de données aux états valide, imposent des restrictions sur les données admises, détection des erreurs (INSERT, UPDATE,DELETE.
I Garantir la cohérence des données.
I Indiquer au système la nature des données.

DECLENCHEURS (dynamique)
Surveille les changements de la base de données, vérifie des conditions et
initie des actions.
Règles *Événement-condition-action*
Quand un *événement* se produit, vérifier la *condition* : si vraie, alors réaliser l'*action*.
Déplacer la logique des applications dans le SGBD.
Garantir les contraintes, Meilleure expressivité
Logique de réparation.

CREATE TRIGGER nom
{ BEFORE | AFTER | INSTEAD OF }
événement
[variables référencées]
[REFERENCING OLD ROW AS X NEW ROW AS Y]
[FOR EACH ROW]
WHEN (condition)
action
OLD ROW AS
NEW ROW AS
OLD TABLE AS
NEW TABLE AS
Classification
Déclenchement (événement) :
Niveau du déclencheur (tuple vs requête).
Moment de l'action.
Condition dans WHEN ou dans l'action.
Non-déterminisme :
Ordre de traitement des tuples.
Plusieurs déclencheurs activés en même temps.
Activation de déclencheurs en chaîne :
Déclenchement récursif et cycle.
Invocation imbriquée.
Variations dans les implémentations (aucun système n'implémente le standard)

INDEX
Améliore performances de BD.
Index = Sélections sur un ensembles d'attributs.
Jointures sur plusieurs tables (création automatique d'index sur clés primaires).

Table de hachage
Accélère sélections avec une contrainte d'égalité
Améliore les performances des jointures.
Coût constant.

Arbre équilibré
Prend en compte l'égalité et les comparaisons (
Accélère la recherche de minimum (MIN) ou le tri de l'affichage (ORDER BY= et coût logarithmique.

Inconvénients
Coût mémoire (marginal) // lors de la création de l'index (medium) //maintenance (peut dépasser gain)
Syntaxe
CREATE UNIQUE INDEX nomIndex ON T(A)
AS DROP INDEX nomIndex

SECURITE
Limiter visibilité et protéger données

GRANT droits ON T TO utilisateurs
[WITH GRANT OPTION]

REVOKE droits ON T FROM utilisateurs
[CASCADE | RESTRICT]
CASCADE Révoque droits attribués transitivement sauf si attribués par un autre
RESTRICT S'arrête si CASCADE aurait révoqué d'autres droits (défaut).

VUE
Syntaxe :

CREATE VIEW **Formations** AS
2 SELECT DISTINCT **formation** FROM **Candidature**

PROCEDURE
Fournit des structures de programmation impérative : procédures, fonctions, instructions de contrôle et variables déclarées.

Utilisation :
Maintenance, renforcer intégrité des données.
Avantages :
Performance : Réduire trafic réseau, plus rapides (compilation possible).
Modularité : découpler gestion des données des applications clientes.
Centraliser la logique avec les données,
Réutilisation portable des procédures.
Sécurité : Gestion des permissions

Syntaxe
BEGIN
instructions
END

Procédure (comparaison avec les fonctions)
Fonction
Peut être utilisée dans n'importe quelle expression.Renvoie un scalaire ou une table.
Procédure
Doit être invoquée dans des instructions séparées.
Peut ne rien renvoyer, renvoyer un ou plusieurs scalaires (en précédant
chaque argument par OUT) ou renvoyer une table.
Structure de contrôle
IF condition THEN
instructions
[ELSIF condition THEN
CASE...
WHEN … THEN...
LOOP... END LOOP
WHILE … DO ...
END WHILE

Curseur
Motivation
Parcourir et manipuler des tables ligne par ligne.
Équivaut au concept d'*itérateur*.
Initialisation et terminaison
OPEN //CLOSE
Lecture des données
FETCH curseur INTO var1 [, var2] …