
Oracle Database 10g : Les fondamentaux du langage SQL (I)

**Volume III • Manuel du stagiaire (Exercices
complémentaires)**

D17108FR10

Edition 1.0

Juin 2004

D39244

ORACLE

Auteur

Nancy Greenberg

Révisions et contributions techniques

Wayne Abbott
Christian Bauwens
Perry Benson
Brian Boxx
Zarko Cesljas
Dairy Chan
Laszlo Czinkocski
Marjolein Dekkers
Matthew Gregory
Stefan Grenstad
Joel Goodman
Rosita Hanoman
Sushma Jagannath
Angelika Krupp
Christopher Lawless
Marcelo Manzano
Isabelle Marchand
Malika Marghadi
Valli Pataballa
Elspeth Payne
Ligia Jasmin Robayo
Bryan Roberts
Helen Robertson
Lata Shivaprasad
John Soltani
Priya Vennapusa
Ken Woolfe

Editeur

Nita K. Brozowski

Copyright © 2004, Oracle. Tous droits réservés.

Cette documentation contient des informations qui sont la propriété d'Oracle Corporation. Fournie sous accord de licence, elle est soumise à des restrictions relatives à l'utilisation et à la publication et est protégée par la loi relative aux droits d'auteur. Il est interdit de rechercher le secret de fabrication du logiciel. Restrictions applicables au gouvernement américain :

If this documentation is delivered to a U.S. Government Agency of the Department of Defense, then it is delivered with Restricted Rights and the following legend is applicable :

Restricted Rights Legend

Use, duplication or disclosure by the Government is subject to restrictions for commercial computer software and shall be deemed to be Restricted Rights software under Federal law, as set forth in subparagraph (c)(1)(ii) of DFARS 252.227-7013, Rights in Technical Data and Computer Software (October 1988).

If this documentation is delivered to a U.S. Government Agency not within the Department of Defense, then it is delivered with "Restricted Rights," as defined in FAR 52.227-14, Rights in Data-General, including Alternate III (June 1987).

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Oracle est une marque déposée, les autres marques Oracle sont des marques d'Oracle Corporation.

Tout autre nom de produit ou de société est utilisé à titre indicatif seulement et peut être une marque de son propriétaire.

Exercices complémentaires

Alioune DIOP (aadiop@hbo-technology.com) has a non-transferable license to use this Student Guide.

Exercices complémentaires

Ces exercices complémentaires peuvent être effectués après l'étude des sujets suivants : instructions SQL SELECT de base, commandes iSQL*Plus de base et fonctions SQL.

1. Le département HR a besoin de déterminer les données relatives à tous les opérateurs embauchés après l'année 1997.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	
143	Randall	Matos	RMATOS	650.121.2874	15-MAR-98	S
144	Peter	Vargas	PVARGAS	650.121.2004	09-JUL-98	S

2. Le département HR a besoin d'un état de tous les employés qui perçoivent une commission. Affichez le nom, le poste, le salaire et la commission de ces employés. Triez les données par ordre décroissant de salaire.

LAST_NAME	JOB_ID	SALARY	COMMISSION_PCT
Abel	SA_REP	11000	.3
Zlotkey	SA_MAN	10500	.2
Taylor	SA_REP	8600	.2
Grant	SA_REP	7000	.15

Exercices complémentaires (suite)

3. Pour des raisons d'évaluation budgétaire, le département HR a besoin d'un état sur les augmentations de salaire prévues. L'état doit afficher les employés qui ne perçoivent aucune commission, mais qui bénéficient d'une augmentation de salaire de 10 % (arrondissez les salaires).

New salary
The salary of King after a 10% raise is 26400
The salary of Kochhar after a 10% raise is 18700
The salary of De Haan after a 10% raise is 18700
The salary of Hunold after a 10% raise is 9900
The salary of Ernst after a 10% raise is 6600
The salary of Lorentz after a 10% raise is 4620
The salary of Mourgous after a 10% raise is 6380
The salary of Rajs after a 10% raise is 3850
The salary of Davies after a 10% raise is 3410
The salary of Matos after a 10% raise is 2860
The salary of Vargas after a 10% raise is 2750
The salary of Whalen after a 10% raise is 4840
The salary of Hartstein after a 10% raise is 14300
The salary of Fay after a 10% raise is 6600
The salary of Higgins after a 10% raise is 13200
The salary of Gietz after a 10% raise is 9130

16 rows selected.

Exercices complémentaires (suite)

- Créez un état des employés et de leur ancienneté. Affichez le nom de tous les employés, ainsi que le nombre d'années et de mois d'ancienneté. Ordonnez l'état en fonction de l'ancienneté. L'employé dont l'ancienneté est la plus importante doit apparaître en haut de la liste.

LAST_NAME	YEARS	MONTHS
King	16	7
Whalen	16	4
Kochhar	14	4
Hunold	14	0
Ernst	12	8

...

Mourgos	4	2
Zlotkey	3	11

20 rows selected.

- Affichez les employés dont le nom commence par la lettre *J*, *K*, *L* ou *M*.

LAST_NAME
King
Kochhar
Lorentz
Matos
Mourgos

- Créez un état qui affiche tous les employés et qui indique par les mots *Yes* ou *No* s'ils perçoivent une commission. Utilisez l'expression `DECODE` dans votre interrogation.

Remarque : La suite des résultats est indiquée page suivante.

LAST_NAME	SALARY	COMMISSIO
King	24000	No
Kochhar	17000	No
De Haan	17000	No
Hunold	9000	No
Ernst	6000	No
Lorentz	4200	No
Mourgos	5800	No
Rajs	3500	No

Exercices complémentaires (suite)

6. (suite)

Davies	3100	No
Matos	2600	No
Vargas	2500	No
Zlotkey	10500	Yes
Abel	11000	Yes
Taylor	8600	Yes
Grant	7000	Yes
Whalen	4400	No
Hartstein	13000	No
Fay	6000	No
Higgins	12000	No
Gietz	8300	No

20 rows selected.

Alioune DIOP (aadiop@hbo-technology.com) has a non-transferable license to use this Student Guide.

Exercices complémentaires (suite)

Ces exercices complémentaires peuvent être effectués après l'étude des sujets suivants : instructions SQL `SELECT` de base, commandes `iSQL*Plus` de base, fonctions SQL, jointures et fonctions de groupe.

7. Créez un état qui affiche le nom du département, le lieu, le nom, le poste et le salaire des employés qui travaillent dans un lieu spécifique. Invitez l'utilisateur à indiquer l'emplacement. Par exemple, si l'utilisateur entre 1800, le résultat suivant s'affiche :

DEPARTMENT_NAME	LOCATION_ID	LAST_NAME	JOB_ID	SALARY
Marketing	1800	Hartstein	MK_MAN	13000
Marketing	1800	Fay	MK_REP	6000

8. Déterminez le nombre d'employés dont le nom se termine par la lettre *n*. Proposez deux solutions.

COUNT(*)
3

9. Créez un état qui affiche le nom, le lieu et le nombre d'employés de chaque département. Assurez-vous que l'état inclut également les départements sans employés.

DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID	COUNT (E.EMPLOYEE_ID)
10	Administration	1700	1
20	Marketing	1800	2
50	Shipping	1500	5
60	IT	1400	3
80	Sales	2500	3
90	Executive	1700	3
110	Accounting	1700	2
190	Contracting	1700	0

8 rows selected.

Exercices complémentaires (suite)

10. Le département HR a besoin de déterminer les postes des départements 10 et 20. Créez un état afin d'afficher les ID de poste de ces départements.

JOB_ID
AD_ASST
MK_MAN
MK_REP

11. Créez un état qui affiche les postes des départements Administration et Executive. Affichez également le nombre d'employés qui occupent ces postes. Affichez en premier le poste occupé par le plus grand nombre d'employés.

JOB_ID	FREQUENCY
AD_VP	2
AD_ASST	1
AD_PRES	1

Alioune DIOP (aadiop@hbo-technology.com) has a non-transferable license to use this Student Guide.

Exercices complémentaires (suite)

Ces exercices complémentaires peuvent être effectués après l'étude des sujets suivants : instructions SQL `SELECT` de base, commandes `iSQL*Plus` de base, fonctions SQL, jointures, fonctions de groupe et sous-interrogations.

12. Affichez tous les employés embauchés au cours de la première quinzaine du mois (avant le 16).

LAST_NAME	HIRE_DATE
De Haan	13-JAN-93
Hunold	03-JAN-90
Lorentz	07-FEB-99
Matos	15-MAR-98
Vargas	09-JUL-98
Abel	11-MAY-96
Higgins	07-JUN-94
Gietz	07-JUN-94

8 rows selected.

13. Créez un état qui affiche les informations suivantes pour tous les employés : nom, salaire, et salaire exprimé en milliers de dollars.

Remarque : La suite des résultats est indiquée page suivante.

LAST_NAME	SALARY	THOUSANDS
King	24000	24
Kochhar	17000	17
De Haan	17000	17
Hunold	9000	9
Ernst	6000	6
Lorentz	4200	4
Mourgos	5800	5
Rajs	3500	3
Davies	3100	3
Matos	2600	2

...

Exercices complémentaires (suite)

13. (suite)

Vargas	2500	2
Zlotkey	10500	10
Abel	11000	11
Taylor	8600	8
Grant	7000	7
Whalen	4400	4
Hartstein	13000	13
Fay	6000	6
Higgins	12000	12
Gietz	8300	8

20 rows selected.

14. Affichez tous les employés dont le manager perçoit un salaire supérieur à 15 000 \$.
Affichez les informations suivantes : nom de l'employé, nom du manager, salaire du manager et niveau de salaire du manager.

LAST_NAME	MANAGER	SALARY	GRADE_LEV
Whalen	Kochhar	17000	E
Higgins	Kochhar	17000	E
Hunold	De Haan	17000	E
Kochhar	King	24000	E
De Haan	King	24000	E
Mourgos	King	24000	E
Zlotkey	King	24000	E
Hartstein	King	24000	E

8 rows selected.

Exercices complémentaires (suite)

15. Affichez le numéro de département, le nom, le numéro des employés et le salaire moyen de tous les départements, ainsi que le nom, le salaire et le poste des employés qui travaillent dans chaque département.

DEPARTMENT_ID	DEPARTMENT_NAME	EMPLOYEES	AVG_SAL	LAST_NAME	SALA
10	Administration	1	4400.00	Whalen	4400
20	Marketing	2	9500.00	Fay	6000
20	Marketing	2	9500.00	Hartstein	13500
50	Shipping	5	3500.00	Rajs	3500
50	Shipping	5	3500.00	Matos	2100
50	Shipping	5	3500.00	Davies	3500
50	Shipping	5	3500.00	Vargas	2100
50	Shipping	5	3500.00	Mourgos	5500
60	IT	3	6400.00	Ernst	6000
60	IT	3	6400.00	Hunold	9000
60	IT	3	6400.00	Lorentz	4800
80	Sales	3	10033.33	Abel	11000
80	Sales	3	10033.33	Taylor	8000
80	Sales	3	10033.33	Zlotkey	10500
90	Executive	3	19333.33	King	24000
90	Executive	3	19333.33	De Haan	17000
90	Executive	3	19333.33	Kochhar	17000
110	Accounting	2	10150.00	Gietz	8000
110	Accounting	2	10150.00	Higgins	12000
190	Contracting	0	No average		

20 rows selected.

16. Créez un état afin d'afficher le numéro de département et le salaire le plus faible du département dont le salaire moyen est le plus élevé.

DEPARTMENT_ID	MIN(SALARY)
90	17000

Exercices complémentaires (suite)

17. Créez un état qui affiche les départements dans lesquels ne travaille aucun commercial. Incluez dans le résultat le numéro de département, le nom du département et le lieu.

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

7 rows selected.

18. Créez les états statistiques suivants pour le département HR : incluez le numéro de département, le nom du département et le numéro des employés qui travaillent dans chaque département qui :

- a. Emploie moins de trois employés :

DEPARTMENT_ID	DEPARTMENT_NAME	COUNT(*)
10	Administration	1
20	Marketing	2
110	Accounting	2

- b. Comporte le plus grand nombre d'employés :

DEPARTMENT_ID	DEPARTMENT_NAME	COUNT(*)
50	Shipping	5

- c. Comporte le plus petit nombre d'employés :

DEPARTMENT_ID	DEPARTMENT_NAME	COUNT(*)
10	Administration	1

Exercices complémentaires (suite)

19. Créez un état qui affiche le numéro d'employé, le nom, le salaire, le numéro de département et le salaire moyen du département, et ce pour tous les employés.

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	AVG(S.SALARY)
100	King	90	19333.3333
200	Whalen	10	4400
101	Kochhar	90	19333.3333
102	De Haan	90	19333.3333
103	Hunold	60	6400
104	Ernst	60	6400
107	Lorentz	60	6400
124	Mourgos	50	3500
141	Rajs	50	3500
142	Davies	50	3500
143	Matos	50	3500
144	Vargas	50	3500
149	Zlotkey	80	10033.3333
174	Abel	80	10033.3333
176	Taylor	80	10033.3333
201	Hartstein	20	9500
202	Fay	20	9500
205	Higgins	110	10150
206	Gietz	110	10150

19 rows selected.

20. Affichez tous les employés qui ont été embauchés le jour de la semaine où le plus grand nombre d'employés ont été embauchés.

LAST_NAME	DAY
Ernst	TUESDAY
Mourgos	TUESDAY
Rajs	TUESDAY
Taylor	TUESDAY
Higgins	TUESDAY
Gietz	TUESDAY

6 rows selected.

Exercices complémentaires (suite)

21. Créez un état anniversaire basé sur la date d'embauche des employés. Triez les anniversaires par ordre croissant.

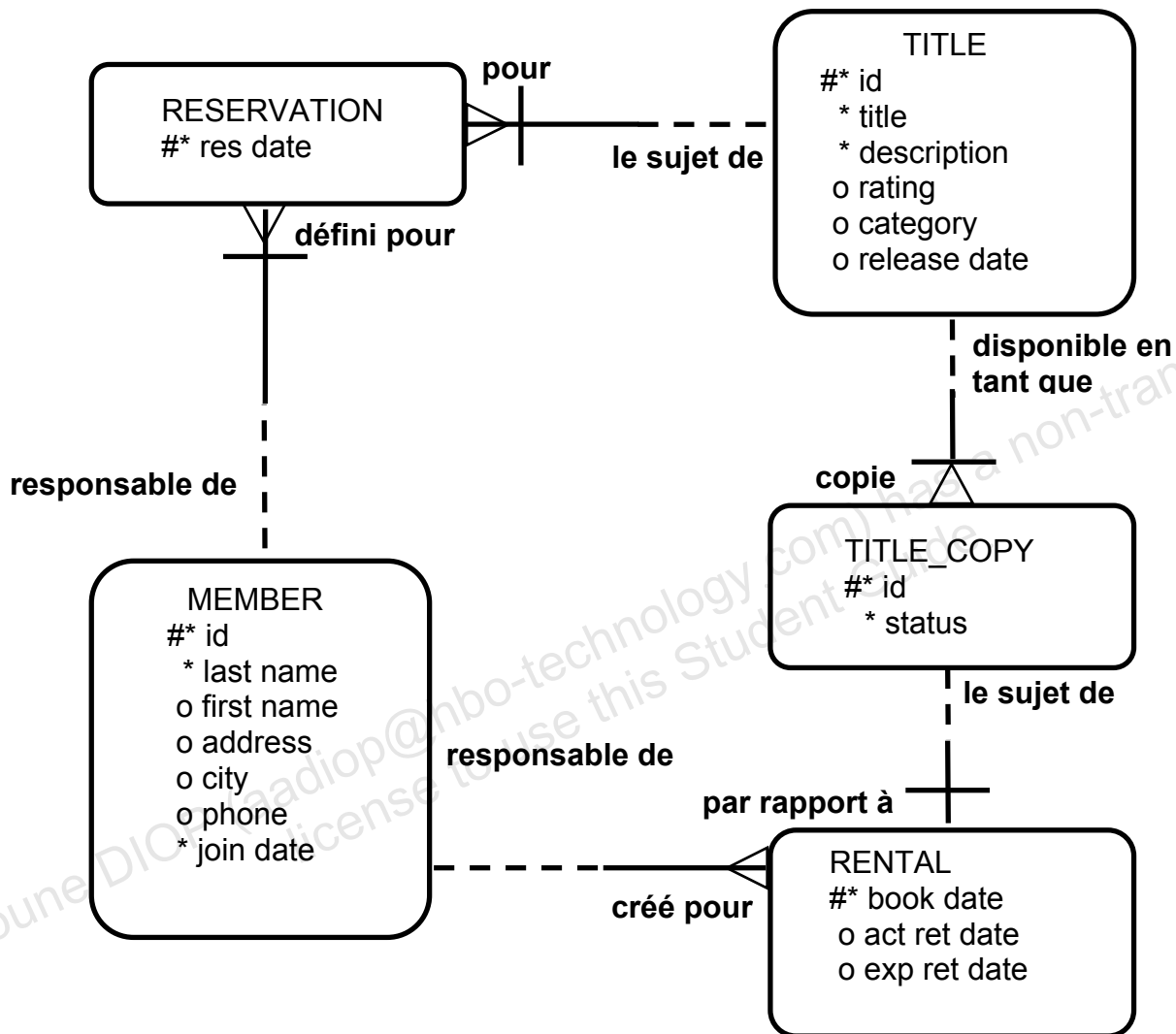
LAST_NAME	BIRTHDAY
Hunold	January 03
De Haan	January 13
Davies	January 29
Zlotkey	January 29
Lorentz	February 07
Hartstein	February 17
Matos	March 15
Taylor	March 24
Abel	May 11
Ernst	May 21
Grant	May 24
Higgins	June 07
Gietz	June 07
King	June 17
Vargas	July 09
Fay	August 17
Whalen	September 17
Kochhar	September 21
Rajs	October 17
Mourgos	November 16

20 rows selected.

Exercices complémentaires : étude de cas

Dans cette étude de cas, vous allez créer un ensemble de tables de base de données pour une application vidéo. Après avoir créé les tables, vous allez insérer, mettre à jour et supprimer des enregistrements dans la base de données d'une boutique vidéo, puis générer un état. La base de données contient uniquement les tables essentielles.

Voici un schéma des entités et attributs de l'application vidéo :



Remarque : Si vous souhaitez créer les tables, vous pouvez exécuter les commandes du script `buildtab.sql`, dans *iSQL*Plus*. Si vous souhaitez supprimer les tables, vous pouvez exécuter les commandes du script `dropvid.sql`, dans *iSQL*Plus*. Vous pouvez ensuite exécuter les commandes du script `buildvid.sql` dans *iSQL*Plus* afin de créer et d'alimenter les tables.

- Si vous utilisez le script `buildtab.sql` pour créer les tables, commencez à l'étape 4.
- Si vous utilisez le script `dropvid.sql` pour supprimer les tables vidéo, commencez à l'étape 1.
- Si vous utilisez le script `buildvid.sql` pour créer et alimenter les tables, commencez à l'étape 6(b).

Exercices complémentaires : étude de cas (suite)

1. Créez les tables à partir des schémas suivants d'instance des tables. Sélectionnez les types de données appropriés et prenez soin d'ajouter des contraintes d'intégrité.

a. Nom de la table : MEMBER

Column_ Name	MEMBER_ ID	LAST_ NAME	FIRST_ NAME	ADDRESS	CITY	PHONE	JOIN_ DATE
Key Type	PK						
Null/ Unique	NN,U	NN					NN
Default Value							System Date
Data Type	NUMBER	VARCHAR2	VARCHAR2	VARCHAR2	VARCHAR2	VARCHAR2	DATE
Length	10	25	25	100	30	15	

b. Nom de la table : TITLE

Column_ Name	TITLE_ ID	TITLE	DESCRIPTION	RATING	CATEGORY	RELEASE_ DATE
Key Type	PK					
Null/ Unique	NN,U	NN	NN			
Check				G, PG, R, NC17, NR	DRAMA, COMEDY, ACTION, CHILD, SCIFI, DOCUMENTARY	
Data Type	NUMBER	VARCHAR2	VARCHAR2	VARCHAR2	VARCHAR2	DATE
Length	10	60	400	4	20	

Exercices complémentaires : étude de cas (suite)

c. Nom de la table : TITLE_COPY

Column Name	COPY_ID	TITLE_ID	STATUS
Key Type	PK	PK,FK	
Null/Unique	NN,U	NN,U	NN
Check			AVAILABLE, DESTROYED, RENTED, RESERVED
FK Ref Table		TITLE	
FK Ref Col		TITLE_ID	
Data Type	NUMBER	NUMBER	VARCHAR2
Length	10	10	15

d. Nom de la table : RENTAL

Column Name	BOOK_DATE	MEMBER_ID	COPY_ID	ACT_RET_DATE	EXP_RET_DATE	TITLE_ID
Key Type	PK	PK,FK1	PK,FK2			PK,FK2
Default Value	System Date				System Date + 2 days	
FK Ref Table		MEMBER	TITLE_COPY			TITLE_COPY
FK Ref Col		MEMBER_ID	COPY_ID			TITLE_ID
Data Type	DATE	NUMBER	NUMBER	DATE	DATE	NUMBER
Length		10	10			10

Exercices complémentaires : étude de cas (suite)

e. Nom de la table : RESERVATION

Column Name	RES_DATE	MEMBER_ID	TITLE_ID
Key Type	PK	PK,FK1	PK,FK2
Null/Unique	NN,U	NN,U	NN
FK Ref Table		MEMBER	TITLE
FK Ref Column		MEMBER_ID	TITLE_ID
Data Type	DATE	NUMBER	NUMBER
Length		10	10

- Examinez le dictionnaire de données afin de vérifier que les tables et les contraintes ont été créées correctement.

TABLE_NAME
MEMBER
RENTAL
RESERVATION
TITLE
TITLE_COPY

CONSTRAINT_NAME	CON	TABLE_NAME
MEMBER_LAST_NAME_NN	C	MEMBER
MEMBER_JOIN_DATE_NN	C	MEMBER
MEMBER_MEMBER_ID_PK	P	MEMBER
RENTAL_BOOK_DATE_COPY_TITLE_PK	P	RENTAL
RENTAL_MEMBER_ID_FK	R	RENTAL

...

COPY_TITLE_ID_PK	P	TITLE_COPY
COPY_TITLE_ID_FK	R	TITLE_COPY

18 rows selected.

Exercices complémentaires : étude de cas (suite)

3. Créez des séquences afin d'identifier de manière unique chaque ligne de la table `MEMBER` et de la table `TITLE`.
 - a. Numérotation des membres de la table `MEMBER` : commencez par 101 ; n'autorisez pas la mise en cache des valeurs. Nommez la séquence `MEMBER_ID_SEQ`.
 - b. Numérotation des titres de la table `TITLE` : commencez par 92 ; n'autorisez pas la mise en cache des valeurs. Nommez la séquence `TITLE_ID_SEQ`.
 - c. Vérifiez l'existence des séquences dans le dictionnaire de données.

SEQUENCE_NAME	INCREMENT_BY	LAST_NUMBER
MEMBER_ID_SEQ	1	101
TITLE_ID_SEQ	1	92

4. Ajoutez des données aux tables. Créez un script pour chaque jeu de données à ajouter.
 - a. Ajoutez les titres des films à la table `TITLE`. Ecrivez un script afin d'insérer les informations relatives aux films.
Enregistrez les instructions dans un script nommé `lab_apcs_4a.sql`. Utilisez les séquences pour identifier chaque titre de manière unique. Entrez les dates de publication au format `DD-MON-YYYY`. Rappelez-vous que les apostrophes des champs de type caractère doivent être gérées de manière spéciale. Vérifiez les ajouts.

TITLE
Willie and Christmas Too
Alien Again
The Glob
My Day Off
Miracles on Ice
Soda Gang

6 rows selected.

Exercices complémentaires : étude de cas (suite)

Title	Description	Rating	Category	Release_date
Willie and Christmas Too	All of Willie's friends make a Christmas list for Santa, but Willie has yet to add his own wish list.	G	CHILD	05-OCT-1995
Alien Again	Yet another installation of science fiction history. Can the heroine save the planet from the alien life form?	R	SCIFI	19-MAY-1995
The Glob	A meteor crashes near a small American town and unleashes carnivorous goo in this classic.	NR	SCIFI	12-AUG-1995
My Day Off	With a little luck and a lot of ingenuity, a teenager skips school for a day in New York.	PG	COMEDY	12-JUL-1995
Miracles on Ice	A six-year-old has doubts about Santa Claus, but she discovers that miracles really do exist.	PG	DRAMA	12-SEP-1995
Soda Gang	After discovering a cache of drugs, a young couple find themselves pitted against a vicious gang.	NR	ACTION	01-JUN-1995

- b. Ajoutez des données à la table MEMBER. Placez les instructions d'insertion dans un script nommé lab_apcs_4b.sql. Exécutez les commandes dans le script. Prenez soin d'utiliser la séquence pour ajouter les numéros des membres.

First_Name	Last_Name	Address	City	Phone	Join_Date
Carmen	Velasquez	283 King Street	Seattle	206-899-6666	08-MAR-1990
LaDoris	Ngao	5 Modrany	Bratislava	586-355-8882	08-MAR-1990
Midori	Nagayama	68 Via Centrale	Sao Paolo	254-852-5764	17-JUN-1991
Mark	Quick-to-See	6921 King Way	Lagos	63-559-7777	07-APR-1990
Audry	Ropeburn	86 Chu Street	Hong Kong	41-559-87	18-JAN-1991
Molly	Urguhart	3035 Laurier	Quebec	418-542-9988	18-JAN-1991

Exercices complémentaires : étude de cas (suite)

c. Ajoutez les exemplaires des films ci-dessous à la table `TITLE_COPY`.

Remarque : Les numéros `TITLE_ID` doivent être disponibles pour cet exercice.

Title	Copy_Id	Status	Title	Copy_Id
Willie and Christmas Too	1	AVAILABLE	Willie and Christmas Too	1
Alien Again	1	AVAILABLE	Alien Again	1
	2	RENTED		2
The Glob	1	AVAILABLE	The Glob	1
My Day Off	1	AVAILABLE	My Day Off	1
	2	AVAILABLE		2
	3	RENTED		3
Miracles on Ice	1	AVAILABLE	Miracles on Ice	1
Soda Gang	1	AVAILABLE	Soda Gang	1

d. Ajoutez les locations ci-dessous à la table `RENTAL`.

Remarque : Le numéro du titre peut être différent selon le numéro de séquence.

Title_Id	Copy_Id	Member_Id	Book_date	Exp_Ret_Date
92	1	101	Il y a 3 jours	Il y a 1 jour
93	2	101	Il y a 1 jour	Dans un jour
95	3	102	Il y a 2 jours	Aujourd'hui
97	1	106	Il y a 4 jours	Il y a 2 jours

Exercices complémentaires : étude de cas (suite)

5. Créez une vue nommée `TITLE_AVAIL` afin d'afficher le titre des films, la disponibilité de chaque copie et la date de retour prévue en cas de location. Interrogez toutes les lignes de la vue. Ordonnez les résultats par titre.

Remarque : Vos résultats peuvent être différents.

TITLE	COPY_ID	STATUS	EXP_RET_DATE
Alien Again	1	AVAILABLE	
Alien Again	2	RENTED	12-FEB-04
Miracles on Ice	1	AVAILABLE	
My Day Off	1	AVAILABLE	
My Day Off	2	AVAILABLE	
My Day Off	3	RENTED	11-FEB-04
Soda Gang	1	AVAILABLE	09-FEB-04
The Glob	1	AVAILABLE	
Willie and Christmas Too	1	AVAILABLE	10-FEB-04

9 rows selected.

6. Apportez des modifications aux données des tables.
- Ajoutez un nouveau titre. Le film est "Interstellar Wars", son classement est PG et sa catégorie est "science fiction". La date de sortie est 07-JUL-77. La description est "Futuristic interstellar action movie. Can the rebels save the humans from the evil empire?". Prenez soin d'ajouter un enregistrement pour deux exemplaires.
 - Entrez deux réservations. Une réservation concerne Carmen Velasquez, qui souhaite louer "Interstellar Wars". L'autre concerne Mark Quick-to-See, qui souhaite louer "Soda Gang".

TITLE	COPY_ID	STATUS	EXP_RET_DATE
Alien Again	1	AVAILABLE	
Alien Again	2	RENTED	12-FEB-04
Interstellar Wars	1	RENTED	10-FEB-04
Interstellar Wars	2	AVAILABLE	
Miracles on Ice	1	AVAILABLE	
My Day Off	1	AVAILABLE	
My Day Off	2	AVAILABLE	
My Day Off	3	RENTED	11-FEB-04
Soda Gang	1	AVAILABLE	09-FEB-04
The Glob	1	AVAILABLE	
Willie and Christmas Too	1	AVAILABLE	10-FEB-04

11 rows selected.

Exercices complémentaires : étude de cas (suite)

7. Apportez une modification à l'une des tables.

- a. Exécutez le script `lab_apcs_7a.sql` afin d'ajouter la colonne `PRICE` à la table `TITLE`, afin d'enregistrer le prix d'achat de la vidéo. Vérifiez vos modifications.

Name	Null?	Type
TITLE_ID	NOT NULL	NUMBER(10)
TITLE	NOT NULL	VARCHAR2(60)
DESCRIPTION	NOT NULL	VARCHAR2(400)
RATING		VARCHAR2(4)
CATEGORY		VARCHAR2(20)
RELEASE_DATE		DATE
PRICE		NUMBER(8,2)

Title	Price
Willie and Christmas Too	25
Alien Again	35
The Glob	35
My Day Off	35
Miracles on Ice	30
Soda Gang	35
Interstellar Wars	29

- b. Créez un script nommé `lab_apcs_7b.sql`, contenant les instructions de mise à jour de chaque vidéo, avec le prix indiqué ci-dessus. Exécutez les commandes du script.

Remarque : Les numéros `TITLE_ID` doivent être disponibles pour cet exercice.

Exercices complémentaires : étude de cas (suite)

8. Créez un état contenant l'historique de chaque client pour la location de vidéos. Prenez soin d'inclure le nom du client, le film loué, les dates de location et la durée de location. Calculez le nombre total de locations pour tous les clients, pour la période indiquée. Enregistrez les commandes qui génèrent l'état dans un fichier script nommé `lab_apcs_8.sql`.

Remarque : Vos résultats peuvent être différents.

MEMBER	TITLE	BOOK_DATE	DURATION
Carmen Velasquez	Alien Again	10-FEB-04	
Carmen Velasquez	Willie and Christmas Too	08-FEB-04	1
LaDoris Ngao	My Day Off	09-FEB-04	
Molly Uguhart	Soda Gang	07-FEB-04	2

Alioune DIOP (aadiop@hbo-technology.com) has a non-transferable license to use this Student Guide.

Alioune DIOP (aadiop@hbo-technology.com) has a non-transferable
license to use this Student Guide.

Exercices complémentaires

Description et données des tables

Alioune DIOP (aadiop@hbo-technology.com) has a non-transferable
license to use this Student Guide.

Table COUNTRIES

DESCRIBE countries

Name	Null?	Type
COUNTRY_ID	NOT NULL	CHAR(2)
COUNTRY_NAME		VARCHAR2(40)
REGION_ID		NUMBER

SELECT * FROM countries;

CO	COUNTRY_NAME	REGION_ID
CA	Canada	2
DE	Germany	1
UK	United Kingdom	1
US	United States of America	2

Table DEPARTMENTS

DESCRIBE departments

Name	Null?	Type
DEPARTMENT_ID	NOT NULL	NUMBER(4)
DEPARTMENT_NAME	NOT NULL	VARCHAR2(30)
MANAGER_ID		NUMBER(6)
LOCATION_ID		NUMBER(4)

SELECT * FROM departments;

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

8 rows selected.

Table EMPLOYEES

DESCRIBE employees

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE
100	Steven	King	SKING	515.123.4567	17-JUN-87
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90
104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-91
107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-99
124	Kevin	Mourgos	KMOURGOS	650.123.5234	16-NOV-99
141	Trenna	Rajs	TRAJS	650.121.8009	17-OCT-95
142	Curtis	Davies	CDAVIES	650.121.2994	29-JAN-97
143	Randall	Matos	RMATOS	650.121.2874	15-MAR-98
144	Peter	Vargas	PVARGAS	650.121.2004	09-JUL-98
149	Eleni	Zlotkey	EZLOTKEY	011.44.1344.429018	29-JAN-00
174	Ellen	Abel	EABEL	011.44.1644.429267	11-MAY-96
176	Jonathon	Taylor	JTAYLOR	011.44.1644.429265	24-MAR-98
178	Kimberely	Grant	KGRANT	011.44.1644.429263	24-MAY-99
200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-87
201	Michael	Hartstein	MHARTSTE	515.123.5555	17-FEB-96
202	Pat	Fay	PFAY	603.123.6666	17-AUG-97
205	Shelley	Higgins	SHIGGINS	515.123.8080	07-JUN-94
206	William	Gietz	WGIEZT	515.123.8181	07-JUN-94

20 rows selected.

Table EMPLOYEES (suite)

JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
AD_PRES	24000			90
AD_VP	17000		100	90
AD_VP	17000		100	90
IT_PROG	9000		102	60
IT_PROG	6000		103	60
IT_PROG	4200		103	60
ST_MAN	5800		100	50
ST_CLERK	3500		124	50
ST_CLERK	3100		124	50
ST_CLERK	2600		124	50
ST_CLERK	2500		124	50
SA_MAN	10500	.2	100	80
SA_REP	11000	.3	149	80
SA_REP	8600	.2	149	80
SA_REP	7000	.15	149	
AD_ASST	4400		101	10
MK_MAN	13000		100	20
MK_REP	6000		201	20
AC_MGR	12000		101	110
AC_ACCOUNT	8300		205	110

20 rows selected.

Table JOBS

DESCRIBE jobs

Name	Null?	Type
JOB_ID	NOT NULL	VARCHAR2(10)
JOB_TITLE	NOT NULL	VARCHAR2(35)
MIN_SALARY		NUMBER(6)
MAX_SALARY		NUMBER(6)

SELECT * FROM jobs;

JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
AD_PRES	President	20000	40000
AD_VP	Administration Vice President	15000	30000
AD_ASST	Administration Assistant	3000	6000
AC_MGR	Accounting Manager	8200	16000
AC_ACCOUNT	Public Accountant	4200	9000
SA_MAN	Sales Manager	10000	20000
SA_REP	Sales Representative	6000	12000
ST_MAN	Stock Manager	5500	8500
ST_CLERK	Stock Clerk	2000	5000
IT_PROG	Programmer	4000	10000
MK_MAN	Marketing Manager	9000	15000
MK_REP	Marketing Representative	4000	9000

12 rows selected.

Table JOB_GRADES

```
DESCRIBE job_grades
```

Name	Null?	Type
GRADE_LEVEL		VARCHAR2(3)
LOWEST_SAL		NUMBER
HIGHEST_SAL		NUMBER

```
SELECT * FROM job_grades;
```

GRA	LOWEST_SAL	HIGHEST_SAL
A	1000	2999
B	3000	5999
C	6000	9999
D	10000	14999
E	15000	24999
F	25000	40000

6 rows selected.

Table JOB_HISTORY

DESCRIBE job_history

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
START_DATE	NOT NULL	DATE
END_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
DEPARTMENT_ID		NUMBER(4)

SELECT * FROM job_history;

EMPLOYEE_ID	START_DATE	END_DATE	JOB_ID	DEPARTMENT_ID
102	13-JAN-93	24-JUL-98	IT_PROG	60
101	21-SEP-89	27-OCT-93	AC_ACCOUNT	110
101	28-OCT-93	15-MAR-97	AC_MGR	110
201	17-FEB-96	19-DEC-99	MK_REP	20
114	24-MAR-98	31-DEC-99	ST_CLERK	50
122	01-JAN-99	31-DEC-99	ST_CLERK	50
200	17-SEP-87	17-JUN-93	AD_ASST	90
176	24-MAR-98	31-DEC-98	SA_REP	80
176	01-JAN-99	31-DEC-99	SA_MAN	80
200	01-JUL-94	31-DEC-98	AC_ACCOUNT	90

10 rows selected.

Table LOCATIONS

DESCRIBE locations

Name	Null?	Type
LOCATION_ID	NOT NULL	NUMBER(4)
STREET_ADDRESS		VARCHAR2(40)
POSTAL_CODE		VARCHAR2(12)
CITY	NOT NULL	VARCHAR2(30)
STATE_PROVINCE		VARCHAR2(25)
COUNTRY_ID		CHAR(2)

SELECT * FROM locations;

LOCATION_ID	STREET_ADDRESS	POSTAL_CODE	CITY	STATE_PROVINCE	CO
1400	2014 Jabberwocky Rd	26192	Southlake	Texas	US
1500	2011 Interiors Blvd	99236	South San Francisco	California	US
1700	2004 Charade Rd	98199	Seattle	Washington	US
1800	460 Bloor St. W.	ON M5S 1X8	Toronto	Ontario	CA
2500	Magdalen Centre, The Oxford Science Park	OX9 9ZB	Oxford	Oxford	UK

Table REGIONS

DESCRIBE regions

Name	Null?	Type
REGION_ID	NOT NULL	NUMBER
REGION_NAME		VARCHAR2(25)

SELECT * FROM regions;

REGION_ID	REGION_NAME
1	Europe
2	Americas
3	Asia
4	Middle East and Africa

Alioune DIOP (aadiop@hbo-technology.com) has a non-transferable license to use this Student Guide.

Exercices complémentaires : solutions

Alioune DIOP (aadiop@hbo-technology.com) has a non-transferable
license to use this Student Guide.

Exercices complémentaires : solutions

Ces exercices complémentaires peuvent être effectués après l'étude des sujets suivants : instructions SQL SELECT de base, commandes iSQL*Plus de base et fonctions SQL.

1. Le département HR a besoin de déterminer les données relatives à tous les opérateurs embauchés après l'année 1997.

```
SELECT *
FROM   employees
WHERE  job_id = 'ST_CLERK'
AND    hire_date > '31-DEC-1997';
```

2. Le département HR a besoin d'un état de tous les employés qui perçoivent une commission. Affichez le nom, le poste, le salaire et la commission de ces employés. Triez les données par ordre décroissant de salaire.

```
SELECT last_name, job_id, salary, commission_pct
FROM   employees
WHERE  commission_pct IS NOT NULL
ORDER BY salary DESC;
```

3. Pour des raisons d'évaluation budgétaire, le département HR a besoin d'un état sur les augmentations de salaire prévues. L'état doit afficher les employés qui ne perçoivent aucune commission, mais qui bénéficient d'une augmentation de salaire de 10 % (arrondissez les salaires).

```
SELECT 'The salary of ' || last_name || ' after a 10% raise is '
      || ROUND(salary*1.10) "New salary"
FROM   employees
WHERE  commission_pct IS NULL;
```

4. Créez un état des employés et de leur ancienneté. Affichez le nom de tous les employés, ainsi que le nombre d'années et de mois d'ancienneté. Ordonnez l'état en fonction de l'ancienneté. L'employé dont l'ancienneté est la plus importante doit apparaître en haut de la liste.

```
SELECT last_name,
       TRUNC(MONTHS_BETWEEN(SYSDATE, hire_date) / 12) YEARS,
       TRUNC(MOD(MONTHS_BETWEEN(SYSDATE, hire_date), 12)) MONTHS
FROM   employees
ORDER BY years DESC, MONTHS desc;
```

Exercices complémentaires : solutions (suite)

5. Affichez les employés dont le nom commence par la lettre *J*, *K*, *L* ou *M*.

```
SELECT last_name
FROM   employees
WHERE  SUBSTR(last_name, 1,1) IN ('J', 'K', 'L', 'M');
```

6. Créez un état qui affiche tous les employés et qui indique par les mots *Yes* ou *No* s'ils perçoivent une commission. Utilisez l'expression `DECODE` dans votre interrogation.

```
SELECT last_name, salary,
       decode(commission_pct, NULL, 'No', 'Yes') commission
FROM   employees;
```

Alioune DIOP (aadiop@hbo-technology.com) has a non-transferable license to use this Student Guide.

Exercices complémentaires : solutions (suite)

Ces exercices complémentaires peuvent être effectués après l'étude des sujets suivants : instructions SQL SELECT de base, commandes iSQL*Plus de base, fonctions SQL, jointures et fonctions de groupe.

7. Créez un état qui affiche le nom du département, le lieu, le nom, le poste et le salaire des employés qui travaillent dans un lieu spécifique. Invitez l'utilisateur à indiquer l'emplacement.

```
SELECT d.department_name, d.location_id, e.last_name, e.job_id,
e.salary
FROM   employees e, departments d
WHERE  e.department_id = d.department_id
AND    d.location_id = &dept_no;
```

8. Déterminez le nombre d'employés dont le nom se termine par la lettre *n*. Proposez deux solutions.

```
SELECT COUNT(*)
FROM   employees
WHERE  last_name LIKE '%n';
--ou
SELECT COUNT(*)
FROM   employees
WHERE  SUBSTR(last_name, -1) = 'n';
```

9. Créez un état qui affiche le nom, le lieu et le nombre d'employés de chaque département. Assurez-vous que l'état inclut également les départements sans employés.

```
SELECT d.department_id, d.department_name,
       d.location_id, COUNT(e.employee_id)
FROM   employees e RIGHT OUTER JOIN departments d
ON     e.department_id = d.department_id
GROUP BY d.department_id, d.department_name, d.location_id;
```

10. Le département HR a besoin de déterminer les postes des départements 10 et 20. Créez un état afin d'afficher les ID de poste de ces départements.

```
SELECT DISTINCT job_id
FROM   employees
WHERE  department_id IN (10, 20);
```

11. Créez un état qui affiche les postes des départements Administration et Executive. Affichez également le nombre d'employés qui occupent ces postes. Affichez en premier le poste occupé par le plus grand nombre d'employés.

```
SELECT e.job_id, count(e.job_id) FREQUENCY
FROM   employees e JOIN departments d
ON     e.department_id = d.department_id
WHERE  d.department_name IN ('Administration', 'Executive')
GROUP BY e.job_id
ORDER BY FREQUENCY DESC;
```


Exercices complémentaires : solutions (suite)

Ces exercices complémentaires peuvent être effectués après l'étude des sujets suivants : instructions SQL SELECT de base, commandes iSQL*Plus de base, fonctions SQL, jointures, fonctions de groupe et sous-interrogations.

12. Affichez tous les employés embauchés au cours de la première quinzaine du mois (avant le 16).

```
SELECT last_name, hire_date
FROM   employees
WHERE  TO_CHAR(hire_date, 'DD') < 16;
```

13. Créez un état qui affiche les informations suivantes pour tous les employés : nom, salaire, et salaire exprimé en milliers de dollars.

```
SELECT last_name, salary, TRUNC(salary, -3)/1000 Thousands
FROM   employees;
```

14. Affichez tous les employés dont le manager perçoit un salaire supérieur à 15 000 \$. Affichez les informations suivantes : nom de l'employé, nom du manager, salaire du manager et niveau de salaire du manager.

```
SELECT e.last_name, m.last_name manager, m.salary, j.grade_level
FROM   employees e JOIN employees m
ON     e.manager_id = m.employee_id
JOIN   job_grades j
ON     m.salary BETWEEN j.lowest_sal AND j.highest_sal
AND    m.salary > 15000;
```

15. Affichez le numéro de département, le nom, le numéro des employés et le salaire moyen de tous les départements, ainsi que le nom, le salaire et le poste des employés qui travaillent dans chaque département.

```
SELECT d.department_id, d.department_name,
       count(e1.employee_id) employees,
       NVL(TO_CHAR(AVG(e1.salary), '99999.99'), 'No average' )
       avg_sal,
       e2.last_name, e2.salary, e2.job_id
FROM   departments d RIGHT OUTER JOIN employees e1
ON     d.department_id = e1.department_id
RIGHT OUTER JOIN employees e2
ON     d.department_id = e2.department_id
GROUP BY d.department_id, d.department_name, e2.last_name, e2.salary,
         e2.job_id
ORDER BY d.department_id, employees;
```

Exercices complémentaires : solutions (suite)

16. Créez un état afin d'afficher le numéro de département et le salaire le plus faible du département dont le salaire moyen est le plus élevé.

```
SELECT department_id, MIN(salary)
FROM   employees
GROUP BY department_id
HAVING AVG(salary) = (SELECT MAX(AVG(salary))
                      FROM   employees
                      GROUP BY department_id);
```

Alioune DIOP (aadiop@hbo-technology.com) has a non-transferable license to use this Student Guide.

Exercices complémentaires : solutions (suite)

17. Créez un état qui affiche les départements dans lesquels ne travaille aucun commercial. Incluez dans le résultat le numéro de département, le nom du département et le lieu.

```
SELECT *
FROM   departments
WHERE  department_id NOT IN (SELECT department_id
                             FROM employees
                             WHERE job_id = 'SA_REP'
                             AND department_id IS NOT NULL);
```

18. Créez les états statistiques suivants pour le département HR : incluez le numéro de département, le nom du département et le numéro des employés qui travaillent dans chaque département qui :

- a. Emploie moins de trois employés :

```
SELECT d.department_id, d.department_name, COUNT(*)
FROM   departments d JOIN employees e
ON     d.department_id = e.department_id
GROUP BY d.department_id, d.department_name
HAVING COUNT(*) < 3;
```

- b. Comporte le plus grand nombre d'employés :

```
SELECT d.department_id, d.department_name, COUNT(*)
FROM   departments d JOIN employees e
ON     d.department_id = e.department_id
GROUP BY d.department_id, d.department_name
HAVING COUNT(*) = (SELECT MAX(COUNT(*))
                   FROM employees
                   GROUP BY department_id);
```

- c. Comporte le plus petit nombre d'employés :

```
SELECT d.department_id, d.department_name, COUNT(*)
FROM   departments d JOIN employees e
ON     d.department_id = e.department_id
GROUP BY d.department_id, d.department_name
HAVING COUNT(*) = (SELECT MIN(COUNT(*))
                   FROM employees
                   GROUP BY department_id);
```

19. Créez un état qui affiche le numéro d'employé, le nom, le salaire, le numéro de département et le salaire moyen du département, et ce pour tous les employés.

```
SELECT e.employee_id, e.last_name, e.department_id, AVG(s.salary)
FROM   employees e JOIN employees s
ON     e.department_id = s.department_id
GROUP BY e.employee_id, e.last_name, e.department_id;
```

Exercices complémentaires : solutions (suite)

20. Affichez tous les employés qui ont été embauchés le jour de la semaine où le plus grand nombre d'employés ont été embauchés.

```
SELECT last_name, TO_CHAR(hire_date, 'DAY') day
FROM   employees
WHERE  TO_CHAR(hire_date, 'Day') =
      (SELECT TO_CHAR(hire_date, 'Day')
       FROM   employees
       GROUP BY TO_CHAR(hire_date, 'Day')
       HAVING COUNT(*) = (SELECT MAX(COUNT(*))
                          FROM   employees
                          GROUP BY TO_CHAR(hire_date, 'Day')));
```

21. Créez un état anniversaire basé sur la date d'embauche des employés. Triez les anniversaires par ordre croissant.

```
SELECT last_name, TO_CHAR(hire_date, 'Month DD') BIRTHDAY
FROM   employees
ORDER BY TO_CHAR(hire_date, 'DDD');
```

Alioune DIOP (aadiop@hbo-technology.com) has a non-transferable license to use this Student Guide.

Exercices complémentaires : solutions de l'étude de cas

1. Créez les tables à partir des schémas suivants d'instance des tables. Sélectionnez les types de données appropriés et prenez soin d'ajouter des contraintes d'intégrité.

a. Nom de la table : MEMBER

```
CREATE TABLE member
(member_id      NUMBER(10)
 CONSTRAINT member_member_id_pk PRIMARY KEY,
 last_name      VARCHAR2(25)
 CONSTRAINT member_last_name_nn NOT NULL,
 first_name     VARCHAR2(25),
 address        VARCHAR2(100),
 city          VARCHAR2(30),
 phone         VARCHAR2(15),
 join_date      DATE DEFAULT SYSDATE
 CONSTRAINT member_join_date_nn NOT NULL);
```

b. Nom de la table : TITLE

```
CREATE TABLE title
(title_id      NUMBER(10)
 CONSTRAINT title_title_id_pk PRIMARY KEY,
 title         VARCHAR2(60)
 CONSTRAINT title_title_nn NOT NULL,
 description   VARCHAR2(400)
 CONSTRAINT title_description_nn NOT NULL,
 rating       VARCHAR2(4)
 CONSTRAINT title_rating_ck CHECK
 (rating IN ('G', 'PG', 'R', 'NC17', 'NR')),
 category     VARCHAR2(20)
 CONSTRAINT title_category_ck CHECK
 (category IN ('DRAMA', 'COMEDY', 'ACTION',
 'CHILD', 'SCIFI', 'DOCUMENTARY')),
 release_date  DATE);
```

c. Nom de la table : TITLE_COPY

```
CREATE TABLE title_copy
(copy_id      NUMBER(10),
 title_id     NUMBER(10)
 CONSTRAINT title_copy_title_id_fk REFERENCES title(title_id),
 status       VARCHAR2(15)
 CONSTRAINT title_copy_status_nn NOT NULL
 CONSTRAINT title_copy_status_ck CHECK (status IN
 ('AVAILABLE', 'DESTROYED', 'RENTED', 'RESERVED')),
 CONSTRAINT title_copy_copy_id_title_id_pk
 PRIMARY KEY (copy_id, title_id));
```

Exercices complémentaires : solutions de l'étude de cas (suite)

d. Nom de la table : RENTAL

```
CREATE TABLE rental
  (book_date      DATE DEFAULT SYSDATE,
   member_id      NUMBER(10)
    CONSTRAINT rental_member_id_fk REFERENCES member(member_id),
   copy_id        NUMBER(10),
   act_ret_date   DATE,
   exp_ret_date   DATE DEFAULT SYSDATE + 2,
   title_id       NUMBER(10),
   CONSTRAINT rental_book_date_copy_title_pk
    PRIMARY KEY (book_date, member_id, copy_id, title_id),
   CONSTRAINT rental_copy_id_title_id_fk
    FOREIGN KEY (copy_id, title_id)
    REFERENCES title_copy(copy_id, title_id));
```

e. Nom de la table : RESERVATION

```
CREATE TABLE reservation
  (res_date       DATE,
   member_id      NUMBER(10)
    CONSTRAINT reservation_member_id REFERENCES member(member_id),
   title_id       NUMBER(10)
    CONSTRAINT reservation_title_id REFERENCES title(title_id),
   CONSTRAINT reservation_resdate_mem_tit_pk PRIMARY KEY
    (res_date, member_id, title_id));
```

2. Examinez le dictionnaire de données afin de vérifier que les tables et les contraintes ont été créées correctement.

```
SELECT  table_name
FROM    user_tables
WHERE   table_name IN ('MEMBER', 'TITLE', 'TITLE_COPY',
                      'RENTAL', 'RESERVATION');

SELECT  constraint_name, constraint_type, table_name
FROM    user_constraints
WHERE   table_name IN ('MEMBER', 'TITLE', 'TITLE_COPY',
                      'RENTAL', 'RESERVATION');
```

Exercices complémentaires : solutions de l'étude de cas (suite)

3. Créez des séquences afin d'identifier de manière unique chaque ligne de la table MEMBER et de la table TITLE.

- a. Numérotation des membres de la table MEMBER : commencez par 101 ; n'autorisez pas la mise en cache des valeurs. Nommez la séquence MEMBER_ID_SEQ.

```
CREATE SEQUENCE member_id_seq  
START WITH 101  
NOCACHE;
```

- b. Numérotation des titres de la table TITLE : commencez par 92 ; n'autorisez pas la mise en cache des valeurs. Nommez la séquence TITLE_ID_SEQ.

```
CREATE SEQUENCE title_id_seq  
START WITH 92  
NOCACHE;
```

- c. Vérifiez l'existence des séquences dans le dictionnaire de données.

```
SELECT sequence_name, increment_by, last_number  
FROM user_sequences  
WHERE sequence_name IN ('MEMBER_ID_SEQ', 'TITLE_ID_SEQ');
```

Exercices complémentaires : solutions de l'étude de cas (suite)

4. Ajoutez des données aux tables. Créez un script pour chaque jeu de données à ajouter.
 - a. Ajoutez les titres des films à la table `TITLE`. Ecrivez un script afin d'insérer les informations relatives aux films.
Enregistrez les instructions dans un script nommé `lab_apcs_4a.sql`. Utilisez les séquences pour identifier chaque titre de manière unique. Entrez les dates de publication au format `DD-MON-YYYY`. Rappelez-vous que les apostrophes des champs de type caractère doivent être gérées de manière spéciale. Vérifiez les ajouts.

```
INSERT INTO title(title_id, title, description, rating,
                  category, release_date)
VALUES (title_id_seq.NEXTVAL, 'Willie and Christmas Too',
        'All of Willie's friends make a Christmas list for
        Santa, but Willie has yet to add his own wish list.',
        'G', 'CHILD', TO_DATE('05-OCT-1995','DD-MON-YYYY'))
/
INSERT INTO title(title_id, title, description, rating,
                  category, release_date)
VALUES (title_id_seq.NEXTVAL, 'Alien Again', 'Yet another
        installment of science fiction history. Can the
        heroine save the planet from the alien life form?',
        'R', 'SCIFI', TO_DATE('19-MAY-1995','DD-MON-YYYY'))
/
INSERT INTO title(title_id, title, description, rating,
                  category, release_date)
VALUES (title_id_seq.NEXTVAL, 'The Glob', 'A meteor crashes
        near a small American town and unleashes carnivorous
        goo in this classic.', 'NR', 'SCIFI',
        TO_DATE('12-AUG-1995','DD-MON-YYYY'))
/
INSERT INTO title(title_id, title, description, rating,
                  category, release_date)
VALUES (title_id_seq.NEXTVAL, 'My Day Off', 'With a little
        luck and a lot ingenuity, a teenager skips school for
        a day in New York.', 'PG', 'COMEDY',
        TO_DATE('12-JUL-1995','DD-MON-YYYY'))
/
...
COMMIT
/
SELECT title
FROM title;
```


Exercices complémentaires : solutions de l'étude de cas (suite)

- b. Ajoutez des données à la table MEMBER. Placez les instructions d'insertion dans un script nommé lab_apcs_4b.sql. Exécutez les commandes dans le script. Prenez soin d'utiliser la séquence pour ajouter les numéros des membres.

```
SET VERIFY OFF
INSERT INTO member(member_id, first_name, last_name,
                  address, city, phone, join_date)
VALUES (member_id_seq.NEXTVAL, '&first_name', '&last_name',
       '&address', '&city', '&phone', TO_DATE('&join_date',
       'DD-MM-YYYY'));
COMMIT;
SET VERIFY ON
```

- c. Ajoutez les exemplaires des films ci-dessous à la table TITLE_COPY.

Remarque : Les numéros TITLE_ID doivent être disponibles pour cet exercice.

```
INSERT INTO title_copy(copy_id, title_id, status)
VALUES (1, 92, 'AVAILABLE')
/
INSERT INTO title_copy(copy_id, title_id, status)
VALUES (1, 93, 'AVAILABLE')
/
INSERT INTO title_copy(copy_id, title_id, status)
VALUES (2, 93, 'RENTED')
/
INSERT INTO title_copy(copy_id, title_id, status)
VALUES (1, 94, 'AVAILABLE')
/;
INSERT INTO title_copy(copy_id, title_id, status)
VALUES (1, 95, 'AVAILABLE')
/
INSERT INTO title_copy(copy_id, title_id, status)
VALUES (2, 95, 'AVAILABLE')
/
INSERT INTO title_copy(copy_id, title_id, status)
VALUES (3, 95, 'RENTED')
/
INSERT INTO title_copy(copy_id, title_id, status)
VALUES (1, 96, 'AVAILABLE')
/
INSERT INTO title_copy(copy_id, title_id, status)
VALUES (1, 97, 'AVAILABLE')
/
```

Exercices complémentaires : solutions de l'étude de cas (suite)

- d. Ajoutez les locations ci-dessous à la table RENTAL.

Remarque : Le numéro du titre peut être différent selon le numéro de séquence.

```
INSERT INTO rental(title_id, copy_id, member_id,
                  book_date, exp_ret_date, act_ret_date)
VALUES (92, 1, 101, sysdate-3, sysdate-1, sysdate-2)
/
INSERT INTO rental(title_id, copy_id, member_id,
                  book_date, exp_ret_date, act_ret_date)
VALUES (93, 2, 101, sysdate-1, sysdate-1, NULL)
/
INSERT INTO rental(title_id, copy_id, member_id,
                  book_date, exp_ret_date, act_ret_date)
VALUES (95, 3, 102, sysdate-2, sysdate, NULL)
/
INSERT INTO rental(title_id, copy_id, member_id,
                  book_date, exp_ret_date, act_ret_date)
VALUES (97, 1, 106, sysdate-4, sysdate-2, sysdate-2)
/
COMMIT
/
```

5. Créez une vue nommée TITLE_AVAIL afin d'afficher le titre des films, la disponibilité de chaque copie et la date de retour prévue en cas de location. Interrogez toutes les lignes de la vue. Ordonnez les résultats par titre.

Remarque : Vos résultats peuvent être différents.

```
CREATE VIEW title_avail AS
SELECT  t.title, c.copy_id, c.status, r.exp_ret_date
FROM    title t JOIN title_copy c
ON      t.title_id = c.title_id
FULL OUTER JOIN rental r
ON      c.copy_id = r.copy_id
AND     c.title_id = r.title_id;

SELECT  *
FROM    title_avail
ORDER BY title, copy_id;
```

Exercices complémentaires : solutions de l'étude de cas (suite)

6. Apportez des modifications aux données des tables.

- a. Ajoutez un nouveau titre. Le film est "Interstellar Wars", son classement est PG et sa catégorie est "science fiction". La date de sortie est 07-JUL-77. La description est "Futuristic interstellar action movie. Can the rebels save the humans from the evil empire?". Prenez soin d'ajouter un enregistrement pour deux exemplaires.

```
INSERT INTO title(title_id, title, description, rating,
                  category, release_date)
VALUES (title_id_seq.NEXTVAL, 'Interstellar Wars',
       'Futuristic interstellar action movie. Can the
       rebels save the humans from the evil empire?',
       'PG', 'SCIFI', '07-JUL-77')
/
INSERT INTO title_copy (copy_id, title_id, status)
VALUES (1, 98, 'AVAILABLE')
/
INSERT INTO title_copy (copy_id, title_id, status)
VALUES (2, 98, 'AVAILABLE')
/
```

- b. Entrez deux réservations. Une réservation concerne Carmen Velasquez, qui souhaite louer "Interstellar Wars". L'autre concerne Mark Quick-to-See, qui souhaite louer "Soda Gang".

```
INSERT INTO reservation (res_date, member_id, title_id)
VALUES (SYSDATE, 101, 98)
/
INSERT INTO reservation (res_date, member_id, title_id)
VALUES (SYSDATE, 104, 97)
/
```

7. Apportez une modification à l'une des tables.

- a. Exécutez le script lab_apcs_7a.sql afin d'ajouter la colonne PRICE à la table TITLE, afin d'enregistrer le prix d'achat de la vidéo. Vérifiez vos modifications.

```
ALTER TABLE title
ADD (price NUMBER(8,2));

DESCRIBE title
```

Exercices complémentaires : solutions de l'étude de cas (suite)

- b. Créez un script nommé `lab_apcs_7b.sql`, contenant les instructions de mise à jour de chaque vidéo, avec le prix indiqué ci-dessus. Exécutez les commandes du script.

Remarque : Les numéros `TITLE_ID` doivent être disponibles pour cet exercice.

```
SET ECHO OFF
SET VERIFY OFF
UPDATE title
SET     price = &price
WHERE  title_id = &title_id;
SET VERIFY OFF
SET ECHO OFF
```

8. Créez un état contenant l'historique de chaque client pour la location de vidéos. Prenez soin d'inclure le nom du client, le film loué, les dates de location et la durée de location. Calculez le nombre total de locations pour tous les clients, pour la période indiquée. Enregistrez les commandes qui génèrent l'état dans un fichier script nommé `lab_apcs_8.sql`.

Remarque : Vos résultats peuvent être différents.

```
SET ECHO OFF
SET VERIFY OFF
SELECT  m.first_name||' '||m.last_name MEMBER, t.title,
        r.book_date, r.act_ret_date - r.book_date DURATION
FROM    member m, title t, rental r
WHERE   r.member_id = m.member_id
AND     r.title_id = t.title_id
ORDER BY member;

SET VERIFY ON
SET ECHO ON
```