

SQL/ECLI

SQL
Executive Command
Language Interface



SOMMAIRE

Historique	3
Base de données et SQL	4
Définition	5
Domaine	6
SQL/400	7
Pourquoi un autre outil SQL	7
Le produit SQL/ECLI	7
Utilisation de SQL/ECLI	10
SQL	10
NAME	11
FILE	11
OPTION	12
OPTIMIZE	12
TEXTE	12
DESCRIPTION DE L'ECRAN SQL	13
Session	13
Largeur	13
Ligne	13
Colonne	13
Fonction	13
En-tête	13
Fichier	13
Message	13
Suite	13
Aide en ligne	13
Synthèse SQL/ECLI	14
Environnement SQL/ECLI	15

Historique

Le nom "SQL" est une abréviation de "*Structured Query Language*" (Langage d'interrogation structuré, en français). C'est un ensemble d'outils permettant de définir et de gérer les données d'une base de données relationnelle.

En 1970, E.F. CODD, membre du laboratoire de recherche d'IBM à San José (Californie) publia un article intitulé "*A Relational Model of Data for Large Shared Data Bank*" (*Communication of the ACM*, vol. 13, n° 6, June 1970). Il y décrivait un ensemble de principes abstraits servant de fondement à un système de gestion des données appelé modèle relationnel. Cet article est à l'origine de toute la technologie des bases de données relationnelles. De nombreuses recherches et expériences tant universitaires qu'industrielles ont donné naissance aux produits relationnels disponibles aujourd'hui sur le marché.

Un aspect particulier des recherches auxquelles nous venons de faire allusion était la conception et l'implantation de prototypes de différents langages relationnels. Il s'agit de langages réalisant, sous une forme syntaxique concrète, toutes ou plusieurs des caractéristiques du modèle relationnel abstrait. Plusieurs langages de ce type ont été créés entre 1970 et 1977. L'un d'eux en particulier s'appelait **SEQUEL** ("*Structured English Query Language*") et avait été défini par D.D. Chamberlin et d'autres chercheurs du laboratoire de recherche d'IBM à San José.

1974	<i>SEQUEL</i>	<i>Implanté sur IBM sous le nom de SEQUEL-XRM.</i>
1976	<i>SEQUEL/2</i>	<i>Version modifiée de SEQUEL. Rebaptisée SQL</i>
1977	<i>System R</i>	<i>généré par SEQUEL/2 (SQL)</i>
1980	<i>ORACLE</i>	<i>(Relational Software Inc.) SQL d'IBM : (Sybase Inc.)</i>
1981	<i>SQL/DS</i>	<i>sous DOS/VSE</i>
1981	<i>INGRES</i>	<i>(Relational Technology Inc.) Interface.</i>
1982	<i>INFORMIX</i>	<i>Informix SQL/SGBD</i>
1982	<i>SQL/DS</i>	<i>sous VM/CMS</i>
1982	<i>X3H2</i>	<i>Proposition de normalisation ANSI</i>
1983	<i>DB2</i>	<i>sous MVS</i>
1984	<i>DG/SQL</i>	<i>Data General</i>
1986	<i>Sybase</i>	<i>(Sybase Inc)</i>
1986	<i>SQL</i>	<i>Ratification ANSI. L'essentiel du dialecte IBM de SQL</i>

Un dernier point d'histoire. La version initiale de SQL était conçue pour une utilisation autonome et interactive; sans doute parce que la normalisation a beaucoup plus d'importance pour la portabilité des programmes que pour les interfaces interactifs, dans l'édification de la norme, l'accent est mis sur l'utilisation de SQL dans des programmes d'accès aux bases de données, plutôt que sur son utilisation en mode interactif.

Les utilisateurs qui n'ont encore aucune connaissance de la gestion des bases de données désireront peut-être consulter au préalable des textes pouvant servir d'introduction à leur étude, tels que "*Bases de Données et Systèmes Relationnels*" (C. DELOBEL, DUNOD, 1992).

Ceux qui souhaitent plus d'informations techniques auront intérêt à consulter "*SQL*" (C.MAREE, Armand COLIN, 1989).

Base de données et SQL

Une base de données est un ensemble d'informations ou de données de même nature. L'annuaire du téléphone, par exemple, est une base de données; ses données sont les noms, adresses et numéros de téléphone des individus ou des sociétés de la ville ou de la région.

Dans l'annuaire téléphonique, les noms de famille sont classés par ordre alphabétique. Si l'on veut appeler Pierrette DURAND, on cherche DURAND, le nom de famille, dans la colonne des noms. Du moment que l'on connaît le nom de famille de la personne à appeler, ce système fonctionne parfaitement.

Mais comment trouver les numéros de téléphone de toutes les personnes qui habitent dans le 11ème arrondissement de Paris ? Comment retrouver le numéro de téléphone de quelqu'un dont on a oublié le nom de famille ? Etant donné que l'annuaire ne comporte pas de classement par arrondissement et par prénoms, il faudrait un temps considérable pour trouver ces numéros.



C'est là qu'intervient SQL.

On peut utiliser SQL pour créer et gérer autant de bases de données que l'on désire. On donne un nom à chaque base de données lors de sa création.

SQL permet de stocker l'information dans des tables. Chaque table n'est rien d'autre qu'un ensemble de rangées et de colonnes.

Une base de données se compose de tables. Une table est un ensemble d'informations qui se présente sous forme de rangées et de colonnes. Une base de données contient au moins une table, mais peut en contenir autant que l'on veut.

Dans une base de données, chaque table contient normalement un type d'informations différent. Par exemple, on aurait probablement des tables distinctes pour les produits vendus, les commandes reçues et les clients servis. Les informations que l'on veut stocker diffèrent selon qu'elles concernent les produits, les commandes ou les clients, d'où la création de tables séparées. On peut ajouter une nouvelle table à n'importe quel moment ou supprimer une table dont on a plus besoin.

Une fois que l'on a stocké des données dans une base, on peut utiliser SQL pour poser des questions concernant ces données. On dit alors que l'on interroge la base de données.

“interroger” signifie rechercher des informations qui sont stockées dans une base de données.

Il existe trois manières différentes d'interroger une base de données :

- on peut rechercher une rangée particulière ou un groupe de rangées
- on peut rechercher une colonne particulière ou un groupe de colonnes
- on peut combiner ces deux techniques pour rechercher des colonnes données situées dans des rangées données.

En outre, on peut utiliser la technique relationnelle appelée lien ou mise en relation (*joining*) pour retrouver des données à partir de plusieurs tables à la fois.

Le terme “*base de données*” est souvent associé à des environnements d'exploitation assez complexes et traitant un volume important de données. En fait, une base de données est simplement un endroit où l'on stocke des informations.

La norme SQL commence par une notion non définie, l'environnement. La définition générale d'environnement dépend de l'implantation du langage; plus exactement, c'est aux auteurs de l'implantation de préciser le sens qu'ils donnent au mot “*environnement*”.

Un environnement particulier ne contient qu'une base de données. On appelle base de données l'ensemble de toutes les données définies par tous les schémas d'un environnement donné. Chaque schéma étant composé d'un ensemble de tables, de vues et de privilèges.

Dans cette optique, les fichiers d'un système d'exploitation peuvent être considérés comme faisant partie intégrante d'une base de données.

Le système de gestion de base de données de l'AS/400 apporte un jeu important de fonctions base de données, et ce qui le rend particulier, elles font partie intégrante du système d'exploitation interne. De ce fait, le système de gestion de base de données est un élément fondamental du système d'exploitation de l'AS/400.

✕ *Les avantages d'un tel système sont majeurs. Par exemple :*

- *les fichiers et les bases de données sont accessibles par une seule interface*
- *tous les programmes qui lisent ou écrivent des données sur l'AS/400 utilisent le même système de gestion de base de données ce qui garantit l'interopabilité des données*
- *l'AS/400 est un ordinateur relationnel, ce qui veut dire que si on prend la peine de concevoir de “bonnes bases de données relationnelles”, la meilleure façon de les exploiter et donc d'accéder rapidement aux données sera d'utiliser SQL, seul, ou avec des langages comme CLP, RPG, C, PL/I, COBOL, etc...*

Définition

SQL est un ensemble d'outils permettant de définir (DDL), de manipuler (DML) et de contrôler (SECURITE) les données d'une base de données relationnelle grâce à un langage assertionnel (QUOI et non COMMENT) mettant en œuvre des fonctions autorisant la sélection, la réunion et la concaténation des colonnes d'une ou plusieurs tables (SELECT, UPDATE, INSERT, DELETE - DML - CREATE, DROP - DDL) et proche du langage naturel (anglais).

Devenu un standard pour l'ANSI (American National Institute, Document ANSI X3.135-1986), l'ISO (International Standards Organization, Document ISO/TC97/SC21/WG3 N117 et N143) et accepté par la proposition X3H2 (American National Standards Database Comittee - Document X3H2-86-61) et SAA IBM.

Il est disponible sur tous les systèmes et toutes les bases de données relationnelles :

<i>MVS</i>	: <i>DB2</i>
<i>DOS/VSE, VM/CMS</i>	: <i>SQL/DS</i>
<i>AS/400</i>	: <i>OS/400</i>
<i>UNIX, DOS, OS2</i>	: <i>INFORMIX, ORACLE, INGRES, SYBASE, ACCESS...</i>

Domaine

Les commandes de “*manipulation*” de données, c'est-à-dire les commandes SQL correspondant aux fonctions de recherche ou de mise à jour des données, peuvent être utilisées de manière interactive (pas toutes) ou intégrées dans un programme d'application (pas toutes).

L'utilisation de la forme interactive signifie que la commande est envoyée d'un terminal auquel le résultat est retourné. Lorsque que la commande est lancée à partir d'un programme d'application, l'opération correspondante fait partie de la tâche d'exécution du programme et le résultat de l'opération est recueilli dans une zone mémoire à l'intérieur du programme.

Un programme écrit dans un langage hôte contenant des instructions SQL est constitué d'un programme hôte classique auquel s'intègre une section de déclarations SQL, un ensemble de définitions de curseurs, un ensemble de déclarations d'exceptions et un ensemble d'instructions SQL.

SQL est un langage d'accès aux bases de données relationnelles et non un langage de programmation générale. La portabilité d'une application de ce type dépend autant de la portabilité du langage hôte que de SQL.

SQL permet la définition des données (CREATE, DROP) par l'intermédiaire de tables (représentation abstraite d'un ensemble de données stockées sur disque) donc intéresse le gestionnaire de base de données et le programmeur.

SQL permet la manipulation des données (SELECT, INSERT, UPDATE et DELETE) donc intéresse le gestionnaire de base de données, le programmeur et l'utilisateur final.

SQL contient trois catégories d'outils autorisant le contrôle des données :

- reprise et contrôle des accès concurrents (gestion des transactions - COMMIT, ROLLBACK)
- intégrité des données (GRANT - NOT NULL - UNIQUE)
- sécurité (Vues - logiques, GRANT, REVOKE - GRTOBJAUT, RVKOBJAUT)

et donc intéresse le responsable des sécurités.



SQL/400

IBM fournit en option, en dehors des produits classiques de manipulation des fichiers, un outil SQL : SQL/400, qui comprend principalement trois parties :

- un “*support run-time SQL*” permettant l'exécution des instructions SQL en faisant l'interface avec les fonctions du système d'exploitation (cette fonctionnalité est intégré au système d'exploitation).
- des *pré-compilateurs SQL* permettant de “*préparer*” un programme source (COBOL/400, PL/I/400, C/400, FORTRAN/400, RPG III - sous ensemble de RPG/400) contenant des instructions SQL
- une *interface interactive* permettant de créer et d'exécuter de manière dynamique (en session) des instructions SQL.

<i>Langages hôtes</i>	<i>AS400</i>	<i>OS/2</i>	<i>370</i>
<i>C</i>	<i>X</i>	<i>X</i>	<i>X</i>
<i>COBOL</i>	<i>X</i>	<i>X</i>	<i>X</i>
<i>FORTRAN</i>	<i>X</i>	<i>O</i>	<i>O</i>
<i>RPG</i>	<i>X</i>	<i>O</i>	<i>O</i>
<i>PLI</i>	<i>X</i>	<i>X</i>	<i>X</i>



Pourquoi un autre outil SQL

Il fallait, à l'aide d'un même produit, pouvoir :

- *trouver portabilité, simplicité, puissance et rapidité de développement en utilisant uniquement des instructions SQL aussi bien avec les langages évolués que les langages de commandes*
- *produire des fichiers ou des états en sortie*
- *conserver l'interface interactive*



Le produit SQL/ECLI

SQL/ECLI met à votre disposition une commande vous permettant d'exécuter une requête SQL/400 natif à partir d'une ligne de commande, dans un programme CL (ce qui n'est pas permis par SQL/400 pour l'instant sauf si on utilise Query Manager trop lent et moins souple) ou dans n'importe quel programme en langage de haut niveau (HLL) et peut être considérée comme un outil de développement.

Cependant, pour s'en servir avec profit, il est nécessaire de connaître la syntaxe de SQL car cette commande ne vous fait bénéficier d'aucune aide syntaxique.

Si le moteur SQL de l'OS400 détecte des erreurs de syntaxe dans votre requête, elles vous seront signalées en messages de premier ou deuxième niveau accessible par F10.

SQL/ECLI utilise pleinement l'implantation SQL d'IBM à un niveau plus simple d'utilisation. Cela a pour conséquence de rendre les utilisateurs beaucoup plus productifs qu'auparavant.



Exemples de programme CLP utilisant des requêtes SQL :

```
PGM          PARM(&AA &MM)
DCL          VAR(&AA) TYPE(*CHAR) LEN(2)
DCL          VAR(&MM) TYPE(*CHAR) LEN(2)
SQL          SQL('select +
              substr(ciel,1,3),sum(nbv),sum(loc1+loc2+loc3+
              +loc4+loc5+loc6),sum(fret1+fret2+fret3+fret4+
              +fret5+fret6),sum(ptt1+ptt2+ptt3+ptt4+ptt5+
              +ptt6) from stataero/gamvcom where aa=' +
              *CAT &AA *CAT ' and mm=' *CAT &MM *CAT +
              ' and cp='C' and nv between 'D' and +
              'K' and ni=2 group by ciel+
              $compa$$mvti$$paxi$$freti$$postei$') +
              FILE(TA1A24) LIBRARY(QTEMP)
SQL          SQL('select +
              tala23.compa,mvtn,paxn,fretn,posten,mvti,paxi,+
              freti,postei from qtemp/tala23,qtemp/tala24 +
              where tala23.compa=tala24.compa') +
              FILE(TA1A2B) LIBRARY(QTEMP)+
SQL          SQL('insert into qtemp/tala2b select +
              tala23.compa,mvtn,paxn,fretn,posten,mvtn-mvtn,+
              paxn-paxn,fretn-fretn,posten-posten +
              from qtemp/tala23 where tala23.compa not +
              in (select compa from qtemp/tala24 where +
              tala24.compa=tala23.compa)')
```


ou encore

```
PGM
DCLF      FILE(EXEMPLE) RCDfmt(*ALL)
DCL       VAR(&SAL) TYPE(*CHAR) LEN(9)

SNDRCVF   RCDfmt(SL)

CHGVAR    VAR(&SAL) VALUE(&SALAIR)
IF        COND(&IN03 *EQ '1') THEN(GOTO CMDLBL(FIN))
SQL       SQL('select * from employes where nom like +
           ''' *CAT &NOM *CAT ''' and srv like ''' +
           *CAT &SRV *CAT ''' and salaire >= ' *CAT &SAL)

FIN:      ENDPGM
```

On peut constater sur ces exemples l'économie de programmation amenée par SQL. Faire la même chose en CLP est pratiquement impossible. En RPG ou PL/1 ou COBOL, le nombre de lignes aurait été multiplié par 20 ou par 100 en fonction des compétences du programmeur.

Si vous utilisez un AGL, SQL reste un complément très utile et efficace.

Utilisation de SQL/ECLI

SQL *SQL('SQL Request')* *NAME(RequestName)* *FILE(FileName)* *LIBRARY(LibraryName)* *OPTIMIZE(Option)* *TEXTE('Text')*

SQL Command Language Interface

Indiquez vos choix, puis appuyez sur ENTREE.

Votre requête >

Nom de requête		Nom	...
Environnement	SUNSQLLEX	Nom	
Fichier de sortie		Nom, *PRINT	
Bibliothèque		Nom, *LIBL, *CURLIB	
Remplacement ou Ajout	*NONE	*NONE, *ADD	
Optimisation	*NONE	*NONE, *TOTAL	
Texte			

Fin

F3=Exit F4=Invite F5=Réafficher F12=Annuler F13=Mode d'emploi invite
F24=Autres touches

Description des paramètres de la commande :

VOTRE REQUETE (paramètre SQL)

Vous devez saisir ici le texte de votre requête SQL/400.

Exemple : *select * from bibliototo where a = 1991*

Si vous spécifiez une fonction d'agrégation dans la clause SELECT, il est possible de donner un nom de colonne à l'expression en la faisant suivre de ce nom encadré par le caractère '\$'.

Exemple : *select count(*)\$TOTAL\$ from bibliototo*

Ce nom remplacera celui généré automatiquement par la commande sous la forme 'FNCxx', où xx est un nombre compris entre 00 et 99.



ATTENTION

L'expression suivante : *select sum(x)\$S1\$, sum(y), sum(w)\$S2\$ from bibliototo*

affectera S1 à sum(x)
 S2 à sum(y)
 FNC00 à sum(w)

NOM DE REQUETE (Paramètre NAME)

Si vous souhaitez utiliser une requête pré-enregistrée, il suffit d'en mentionner ici le nom.

ENVIRONNEMENT (Paramètre NAME)

Si vous utilisez un nom de requête pré-enregistré, indiquez ici le nom de la bibliothèque dans laquelle elle se trouve.

FICHER DE SORTIE (Paramètre FILE)

Si vous souhaitez que le résultat de votre requête soit stocké dans un fichier de sortie, indiquez ici le nom du fichier à créer.

Il est possible de spécifier un ou plusieurs index sur le nom d'une colonne en terminant ce dernier par le caractère "£". On obtient bien sûr un fichier trié mais surtout indexé au sens de l'OS/400.

Exemple :

```
select substr(commune,1,5)$CODPOST£$ from biblio/toto FILE(TEST) LIBRARY(QTEMP)
```

Lorsque vous utilisez '*PRINT' comme nom de fichier, un état sommaire est généré.

BIBLIOTHEQUE (Paramètre FILE)

Si vous souhaitez que le résultat de votre requête soit stocké dans un fichier de sortie, indiquez ici le nom de la bibliothèque dans laquelle le fichier devra être placé.

REPLACEMENT OU AJOUT (Paramètre OPTION)

Si vous souhaitez que le résultat de votre requête soit stocké à la suite des données existantes dans le fichier de sortie, indiquez l'option *ADD. Par défaut, les données existantes sont remplacées.



ATTENTION

S'il s'agit d'un fichier existant, il doit avoir le même format d'enregistrement que celui produit par votre instruction SQL. Si le fichier n'existe pas, SQL/ECLI le crée automatiquement.

OPTIMISATION (Paramètre OPTIMIZE)

Si le nombre d'enregistrements répondant à votre requête n'est pas trop grand, vous pouvez indiquer ici *TOTAL. Dans ce cas, la commande extraira l'ensemble des réponses en une seule fois et vous affichera le nombre de réponses dès le départ.

TEXTE (Paramètre TEXTE)

Documentez cette zone si vous souhaitez associer un texte à votre fichier de sortie ou spécifier un titre à votre état imprimé.

Exemple de requête :

```

                                SQL Command Language Interface

Indiquez vos choix, puis appuyez sur ENTREE.

Votre requête . . . . . > select * from employes

Nom de requête . . . . .                               Nom
Environnement . . . . .                               SUNSQLEX      Nom
Fichier de sortie . . . . .                           Nom, *PRINT
Bibliothèque . . . . .                               Nom, *LIBL, *CURLIB
Remplacement ou Ajout . . . . .                       *NONE        *NONE, *ADD
Optimisation . . . . .                               *NONE        *NONE, *TOTAL
Texte . . . . .

                                                                Fin
F3=Exit   F4=Invite   F5=Réafficher   F12=Annuler   F13=Mode d'emploi invite
F24=Autres touches

```

```

*LIBL/EMPLOYES                Affichage des données (ECLI)    16/01/94  11:58:26
                                Largeur des données . . . : 49
Première ligne à afficher . . . Première colonne à afficher .
MAT  NOM      SRV  POSTE  SX SALAIR  PRIME
10  CHRISTIAN A00 DIRECT GEN M  45000.00  0.00
20  MICHEL    B01 CHEF  SERV  M  22000.00 1000.00
30  JEANNE    C01 CHEF  SERV  F  26000.00  0.00
50  GEORGES   E01 CHEF  SERV  M  32000.00  0.00
60  IRENE     D11 CHEF  SERV  F  28000.00  600.00
70  EVE       D21 CHEF  SERV  F  19000.00  0.00
90  HELENE    E11 CHEF  SERV  F  29750.00  600.00
100 THEODORE  D01 CHEF  SERV  M  32000.00 1500.00
110 VINCENT  A00 DIRECT COM M  36500.00 9000.00
120 JEAN     A00 COMPTABLE M  15000.00  600.00
130 DOLORES  C01 ANALYSTE F  13800.00  500.00
140 CLAUDE   C01 ANALYSTE M  13900.00  600.00
150 BERNARD  D11 PROGRAMMR M   8000.00 1500.00
160 ELISABETH D11 PROGRAMMR F   8500.00  400.00
170 PMASATOSHI D11 PROGRAMMR M   8675.00  500.00
180 MARILYNE D11 PROGRAMMR F   9500.00  0.00

                                                                A Suivre...
F3=Quitter F11=Zones F19=Gauche F20=Droite F21=Commande Système F24=Autres
Requête terminée. 33 ligne(s) trouvée(s).

```

DESCRIPTION DE L'ECRAN SQL

Session	Cet écran vous permet de visualiser le résultat de votre requête. Vous pouvez faire le faire défiler s'il ne contient pas toute l'information.
Largeur	Longueur totale de la ligne affichée.
Ligne	Cette zone vous permet de saisir le numéro de la ligne que vous souhaitez visualiser. Il ne doit pas être supérieur au nombre total de lignes déjà lues affiché au bas de l'écran. Si toutes les lignes ne sont pas encore disponibles, faites défiler l'écran pour en effectuer la lecture.
Colonne	Cette zone vous permet de saisir le numéro de la colonne à partir de laquelle vous souhaitez l'affichage. Il ne doit pas être supérieur à la longueur maximale de la ligne. Si toutes les colonnes, ou celles souhaitées, ne sont pas visibles, faites défiler les informations à l'aide des touches de fonction préconisées.
Fonction	Les différentes touches de fonctions vous permettent de déplacer l'affichage (si possible), d'exécuter une commande système ou de quitter l'application.
En-tête	Cette ligne affiche les noms des colonnes du fichier.
Fichier	Il s'agit soit du nom du fichier ou de la table concernée par la requête dans le cas d'un 'SELECT * FROM nom_de_fichier' soit du nom du fichier temporaire créé par la commande (PFxxxxxxxx).
Message	Messages d'erreur ou d'information.
Suite	Indicateur de limite de défilement.
Aide en ligne	A tout moment, l'appui de la touche "aide" provoque l'affichage d'une information contextuelle.

Synthèse SQL/ECLI

SQL/ECLI est utilisé pour :

- Visualiser le résultat d'une requête SQL
- Enregistrer le résultat d'une requête SQL dans un fichier
- Imprimer le résultat d'une requête SQL
- Exécuter toute instruction SQL

SQL/ECLI peut être lancé à partir de la ligne de commande, en interactif, d'un source CL ou de n'importe quel programme en langage de haut niveau.

SQL SQL('SQL Request') NAME(RequestName) FILE(FileName) LIBRARY(LibraryName) OPTIMIZE(Option) TEXT('Text')

Instructions

Toutes les "instructions" SQL sont autorisées, y compris les instructions autres que SELECT, telles que :

COMMENT ON	COMMIT	CREATE COLLECTION
CREATE INDEX	CREATE TABLE	CREATE VIEW
DELETE	DROP	GRANT
INSERT	LABEL ON	LOCK TABLE
REVOKE	ROLLBACK	UPDATE
...		

Colonnes

Les colonnes obtenues par les fonctions

AVG, COUNT, MAX, MIN, SUM, ...

ou par les fonctions scalaires

DECIMAL, DIGITS, FLOAT, INTEGER, LENGTH, SUBSTR, ...

ou par les opérateurs

!!, +, -, /, *

peuvent être renommées comme suit

\$NOM\$

n'importe où à l'intérieur de la phrase.

Options

Différentes options apportent d'autres fonctionnalités à SQL/ECLI :

- *TOTAL** Permet de forcer l'affichage du résultat après l'exécution de la requête
- *ADD** Permet d'ajouter le résultat de la requête à la fin du fichier de sortie
- *PRINT** Permet de générer un état imprimé à partir du résultat de la requête

- **Remarque :** *La commande SQL de SQL/ECLI peut être appelée, en interactif, de façon réursive.*



Environnement SQL/ECLI

La commande SQL de **SQL/ECLI** vous permet de stocker votre requête (F16) après mise au point ou de créer automatiquement un programme CL utilisant votre requête (F18) sans avoir à la resaisir.

SQL/ECLI met aussi à votre disposition la commande **SQLLIB** vous permettant de gérer vos requêtes :

- Rappel d'une requête existante et exécution
- Visualisation de la syntaxe SQL d'une requête existante
- Edition du source CL contenant votre requête
- Modification d'une requête existante
- Modification d'une description de requête
- Copie d'une requête
- Suppression d'une requête
- Entrée sous PDM
- Lancement d'une commande système

```

                                Entrée de commandes
                                Niveau de demande: 4

Commandes et messages précédents :
> s .....
F :   SQL/ECLI      Vos requêtes SQL AS/400      V4R9 :
> e :
   :   Environnement [ SUNSQLEX ]      A partir de . . :
   :   E=Edit_source  M=Modifier_Requête  S=Voir_SQL  V=Vue_Requête :
   :   X=Exécuter   2=Texte   3=Copier   4=Supprimer :
   :
   :   NOM      Description :
   :   A_EMPLOYES  Vue du fichier EMPLOYES :
   :   A_PROJETS   Vue du fichier PROJETS :
   :   A_SERVICES  Vue du fichier SERVICES :
   :   B_EX1      Nom des employés et nom des services :
   :   B_EX2      Employés du service 'D01' : n
Tapez :   B_EX3      Total des salaires pour le service 'D01' :
====> s :   B_EX4      Retrouve tous les projets utilisant BERNARD :+
   :
   :   F3=Exit   F5=Réafficher   F6=Créer   F8=PDM   F24=Autres touches :
   :
F3=Exi :
F11=Pl :.....

```

Auparavant vous utilisiez CLP, RPG, COBOL ou C, avec peut-être QUERY.

Avec **SQL/ECLI** vous améliorerez votre productivité de manière conséquente. Si vous utilisez un AGL, vous constaterez très vite ce que peut vous apporter SQL.

Si vous avez déjà SQL/400 d'IBM, **SQL/ECLI** (pour un investissement modique) vous donne un complément efficace (stockage et réutilisation des requêtes, appel de requêtes depuis CLP,...).