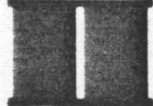


dBASE™



Ashton-Tate

QUESTIONS ET RÉPONSES

Numéro 1 - Février 1984

Sommaire

I - AVANT D'UTILISER dBASE II

- 1.1. Configuration minimale pour l'utilisation de dBASE II
- 1.2. Capacité de la mémoire centrale
- 1.3. Installation de dBASE II pour le système d'exploitation MP/M
- 1.4. Equivalence des caractères
- 1.5. dBASE II et un disque dur ?
- 1.6. Calcul de la taille d'un fichier de données dBASE II
- 1.7. Evaluation de la taille d'un fichier INDEX (.NDX) avant sa création

II - UTILISATION DE dBASE II

- 2.1. Modification d'un fichier de REPORT (.FRM)
- 2.2. Utilisation des fonctions spéciales d'une imprimante
- 2.3. Impression de la dernière ligne
- 2.4. Suppression du saut de page
- 2.5. Utilisation de routines en langage machine
- 2.6. Génération d'état sélectif sur des fichiers volumineux
- 2.7. Possibilités particulières sur l'IBM-PC, le VICTOR, le TEXAS-PC avec la version 2.4 de dBASE II
- 2.8. Sélection des attributs vidéo ou de la couleur

III - dBASE II, Version 2.4

- 3.1. Corrections de la version 2.4

IV - PROBLÈMES ET SOLUTIONS

- 4.1. Perte d'enregistrements
- 4.2. Utilisation d'un autre éditeur de texte que MODIFY COMMAND
- 4.3. "FICHIER EN COURS D'UTILISATION"
- 4.4. Perturbation de l'affichage pendant la saisie
- 4.5. "ENREGISTREMENT NON SITUE"
- 4.6. Instruction FILE(<nom de fichier>)
- 4.7. JOIN
- 4.8. MODIFY COMMAND
- 4.9. Utilisation de l'instruction USING
- 4.10. Saisie de caractères codés sur 8 bits avec MODIFY COMMAND

V - UN SUPPORT TECHNIQUE TÉLÉPHONIQUE A VOTRE DISPOSITION



La Commande Electronique

7, RUE DES PRIAS — 27920 SAINT-PIERRE DE BAILLEUL
TÉL. (32) 52.54.02 TÉLEX LCE 180 855

1 - AVANT D'UTILISER dBASE II

Q. 1.1. Quelle est la configuration minimale pour l'utilisation de dBASE II ?

R. Micro-ordinateur 8 bits

- micro-ordinateur 8080, 8085 ou Z80
- système d'exploitation CP/M (version 2.x)
- mémoire centrale de 48 K-octets disponibles (Apple 56 K).

Micro-ordinateur 16 bits

- micro-ordinateur 8088 ou 8085
- système d'exploitation MS-DOS (PC-DOS) ou CP/M-86
- mémoire centrale de 92 K-octets disponibles.

Q. 1.2. dBASE II utilise-t-il la mémoire centrale supérieure au minimum nécessaire ?

R. Non dBASE II utilise une taille mémoire fixe sans tenir compte de la taille mémoire disponible. dBASE utilise la mémoire jusqu'à l'adresse A400 Hexadécimale (41984 décimale) sur les micro-ordinateurs 8 bits et jusqu'à l'adresse DD98 Hexadécimale (56728 décimale) sur les micro-ordinateurs 16 bits. Ces deux valeurs correspondent au minimum nécessaire pour tous les traitements de dBASE II sauf pour la commande SORT.

Sur les systèmes 8 bits, la commande SORT utilise la mémoire depuis l'adresse A400H jusqu'à l'adresse de départ du BDOS, et sur les systèmes 16 bits, elle utilise la mémoire depuis l'adresse DD98H jusqu'à la limite de 92 K.

Q. 1.3. Comment installer dBASE II pour le système d'exploitation MP/M ?

R. Pour permettre un fonctionnement correct de dBASE II sous MP/M, vous devez effectuer les étapes suivantes :

- Exécuter le programme d'installation (A>INSTALL)
- Sélectionner votre terminal
- Répondre Y à la question : CHANGE MACRO, DATE, ETC ?
- Sélectionner le système d'exploitation MP/M
- Sauvegarder l'installation.

Les modifications effectuées dans le programme DBASE.COM forceront le système d'exploitation à fermer vos fichiers lorsque vous les fermez sous dBASE II.

Q. 1.4. Equivalence des caractères

R. En raison des caractéristiques de certains claviers, la documentation de dBASE II en français laisse apparaître quelques ambiguïtés, les caractères figurant dans la documentation correspondent à ceux du clavier d'un micro-ordinateur AZERTY accentué codé sur 7 bits. Vous trouverez ci-dessous le tableau d'équivalence des caractères utilisés dans la documentation avec ceux d'un micro-ordinateur équipé d'un clavier AZERTY ou d'un clavier accentué codé sur 8 bits (IBM-PC, etc.).

| Caractères de la documentation | Caractères équivalents | Valeur hexadécimale |
|--------------------------------|------------------------|---------------------|
| à | | 40 |
| £ | | 23 |
| ° | | 5B |
| § | | 5D |

Q. 1.5. dBASE II peut-il utiliser un disque dur ?

R. Oui, dBASE II peut utiliser un disque dur. Les systèmes d'exploitation CP/M-80 et CP/M-86 limitent la taille d'un fichier à 8 Méga-octets. MS-DOS (PC-DOS) la limite à 32 Méga-octets. La version actuelle de dBASE II limite la taille d'un fichier de données à 8 Méga-octets aussi bien sous CP/M-80, CP/M-86 ou MS-DOS. La version 2.0 de MS-DOS vous permet d'utiliser des sous-directory mais tous les fichiers utilisés par dBASE II doivent se trouver dans le même directory ou sous-directory.

Q. 1.6. Comment calculer la taille d'un fichier de données dBASE II ?

- R. Description d'un fichier de données dBASE II
1. La structure (nom et définition de chaque rubrique, date, compteur d'enregistrements...) utilise 521 octets (caractères).
 2. Tous les enregistrements ont un octet supplémentaire par enregistrement (utilisé pour positionner le pointeur d'effacement).

Taille du fichier = (nb d'enregistrements x (longueur d'un enregistrement + 1)) + 521

Q. 1.7. Est-il possible d'évaluer la taille d'un fichier INDEX (.NDX) avant sa création ?

- R. La taille approximative d'un fichier index peut être calculée de la façon suivante :

CLE = Nombre de caractères de la clé

ENR = Nombre d'enregistrements

Taille d'un fichier INDEX = 1,4 (INT(ENR/INT(512/(CLE+4))))x512

Le résultat sera exprimé en nombre de caractères (octets).

II - UTILISATION DE dBASE II

Q. 2.1. Est-il possible de modifier un fichier de REPORT (.FRM), ou est-il nécessaire de le recréer ?

- R. Vous pouvez modifier un fichier .FRM en utilisant **MODIFY COMMAND <nom de fichier>.FRM**. Vous verrez alors apparaître toutes les réponses que vous avez fournies lors de la création. Il est conseillé lors de la création de faire une copie d'écran sur l'imprimante afin de disposer, au moment de la modification, de toute la définition sous une forme claire. Nous tenons à vous préciser qu'aucun contrôle de syntaxe ne pourra être effectué lors de la sauvegarde (il en est de même pour les fichiers de commande) et que vous devez procéder avec prudence pour toutes les modifications. Avant de pouvoir modifier le contenu d'un fichier .FRM, il est nécessaire que ce dernier soit fermé. Un fichier de REPORT est fermé lorsque le fichier de données qui lui est associé est fermé (USE (CR)).

Q. 2.2. Est-il possible d'utiliser les fonctions spéciales d'une imprimante (ESCAPE + chaîne de caractères) avec dBASE II ?

- R. dBASE II vous permet d'envoyer n'importe quelle chaîne de caractères vers l'imprimante, à l'exclusion du caractère dont la valeur binaire est zéro. Par exemple, supposons que la chaîne de caractères pour sélectionner le mode compressé de votre imprimante soit :

ESCAPE + N + 3

vous utiliserez la procédure suivante :

SET PRINT ON

?? CHR(27)+"N3"

Nous vous rappelons que la fonction **CHR(valeur décimale)** de dBASE II (et non pas **CHR\$(valeur décimale)** de BASIC) vous permet d'envoyer vers l'écran ou vers l'imprimante, le caractère ASCII correspondant. (**CHR(65)** est identique au caractère A). La fonction **CHR** peut également être utilisée avec une commande **à...SAY**.

Q. 2.3. Pour quelle raison, lors d'une édition sur l'imprimante, la dernière ligne n'est imprimée que lors du lancement d'une nouvelle impression ou lors de la sortie (QUIT) de dBASE II ?

- R. La plupart des imprimantes sont dotées d'une mémoire-tampon (buffer) et n'imprime une ligne que lorsqu'elles ont reçu les codes ODH-OAH (carriage return - line feed).
- Si vous êtes en mode **SET PRINT ON**, placez **?? CHR(13)** à la fin de votre fichier de commande d'impression ou sur la ligne suivant la commande de génération d'état (**REPORT ... TO PRINT**).
 - Si vous êtes en mode **SET FORMAT TO PRINT**, placez à <coordonnée> **SAY CHR(13)** à la fin de votre format d'impression.

2.4. Suppression du saut de page

Si la commande **SET EJECT OFF** est émise avant l'impression d'un état (**REPORT**), seul le premier saut de page avant le début de l'impression sera supprimé. Pour supprimer l'émission du code de fin de page pendant l'édition de l'état, vous devrez utiliser certaines modifications en mémoire centrale (**POKE**). dBASE exécute sa routine d'émission du code de saut de page si :

- vous utilisez la commande **EJECT**
- si les coordonnées de **à<ligne>,<colonne>** représentent une position antérieure aux dernières coordonnées émises. (**à 10,0 SAY...**) émise après (**à 4,0 SAY...**).

Lors de l'émission du code de saut de page, dBASE :

- enverra vers l'imprimante le code de saut de page (**OCH**)
- remettra à 0 le compteur de lignes.

Vous trouverez ci-dessous les instructions qui vous permettent de supprimer et de réaliser l'émission du saut de page, dans l'un des cas cités ci-dessus.

Pour supprimer l'émission du saut de page :

| | |
|------------------------|-------------------------|
| POKE 15021,0,0 | (version 2.3B, 8 bits) |
| POKE 15170,0,0 | (version 2.4, 8 bits) |
| POKE 19329,144,144,144 | (version 2.3D, 16 bits) |
| POKE 20100,144,144,144 | (version 2.4, CP/M-86) |
| POKE 20034,144,144,144 | (version 2.4, MS-DOS) |

Pour réactiver l'émission du saut de page :

| | |
|----------------------------|-------------------------|
| POKE 15021,205,0 | (version 2.3B, 8 bits) |
| POKE 15170,205,0 | (version 2.4, 8 bits) |
| POKE 19329,232,197,183 | (version 2.3D, 16 bits) |
| POKE 20100,232,232,203,180 | (version 2.4, CP/M-86) |
| POKE 20034,232,205,181 | (version 2.4, MS-DOS) |

2.5. Utilisation de routines en langage machine

Vous trouverez ci-dessous les syntaxes d'utilisation de routines en langage machine. Ces descriptifs ne figurent pas dans l'actuelle documentation.

1) SET CALL TO <adresse>

définit l'adresse de départ de la routine à exécuter. Cette adresse exprimée en décimale devra être postérieure à :

- 41984 pour les versions 8 bits
- 56728 pour les versions 16 bits.

NB Nous vous rappelons que, lors de l'utilisation de la commande **SORT**, les adresses de 41984 à l'adresse de départ du BDOS pour les versions 8 bits et de 56728 jusqu'à la limite de 92 K sont utilisées, ce qui entraîne une destruction d'une éventuelle routine préalablement positionnée.

2) CALL (<variable mémoire>)

exécute une routine en langage machine à partir de l'adresse définie par **SET CALL TO <adresse>**. Une chaîne de caractères (<variable mémoire>) peut être transmise pour l'exécution de la routine (254 octets sont disponibles). Le retour à dBASE se fera dès la rencontre de l'instruction **SET** dans la routine.

3) LOAD <fichier>

vous permet de charger en mémoire un fichier hexadécimal (**HEX** pour 8 bits, **H86** pour 16 bits) au format hexadécimal INTEL.

4) POKE <adresse>,<octet1>,<octet2>,...

place 1 ou les octets cités (exprimés en décimale) à l'<adresse> spécifiée.

5) PEEK (<adresse>)

retourne la valeur décimale contenue à l'<adresse> de la mémoire.

2.6. Génération d'état sélectif sur des fichiers volumineux

Utilisation de la commande REPORT

Avantage :

Elle permet une mise en oeuvre rapide d'une génération d'état avec la possibilité de total et de sous-total.

Inconvénient :

La génération de l'état s'effectue séquentiellement sur l'ensemble du fichier de données avec éventuellement une organisation liée à un fichier index.

Si une condition est donnée, l'ensemble du fichier de données sera lu et seul(s) le(s) enregistrement(s) répondant au(x) critère(s) de sélection sera(ont) imprimé(s), ce qui entraîne un délai de génération proportionnel à la taille du fichier de données.

Solution :

Pour obtenir rapidement un état sélectif sur la clé d'un gros fichier indexé, il est préférable d'utiliser le principe suivant :

| | |
|--|--|
| ACCEPT TO PARAMETRE | |
| FIND &PARAMETRE | Recherche de la clé d'impression. |
| IF £=0 (aucun enregistrement ne correspond à la clé recherchée) | A ce moment, vous vous trouvez sur le premier enregistrement correspondant à votre recherche. |
| ELSE | |
| SET PRINT ON (par exemple) | |
| STORE 0 TO LIGNE | |
| DO WHILE "&PARAMETRE" (\$) VARIABLE .AND. .NOT. EOF | |
| ? Variable,Variable,etc | (Variable = champ de clé) Vous pouvez également utiliser un fichier format. |
| STORE LIGNE+1 TO LIGNE | |
| IF LIGNE > 50 | |
| EJECT | Saut de page après 50 lignes par exemple |
| STORE 0 TO LIGNE | |
| ENDIF | |
| SKIP+1 | |
| ENDDO | |
| ENDIF | |

2.7. Possibilités particulières sur l'IBM-PC, le VICTOR, le TEXAS-PC avec la version 2.4 de dBASE II

Programmation des touches de fonction

Il vous est maintenant possible de programmer chacune des 10 touches de fonction (F1 à F10) du clavier de l'IBM-PC.

La syntaxe de programmation est la suivante :

SET Fn TO "chaîne de caractères"

Ex. : **SET F3 TO "LIST FOR"**

Lors de l'appui sur la touche F3, **LIST FOR** sera affiché et vous pourrez compléter la condition.

Vous pourrez également inclure un retour chariot à la fin de la chaîne de caractères par l'intermédiaire du caractère ";".

Ex. : **SET F4 TO "LIST ;"**

L'appui sur la touche F4 entraînera l'émission de la commande **LIST** suivie d'un retour chariot.

Les affectations par défaut sont les suivantes :

| | |
|----|-----|
| F1 | F6 |
| F2 | F7 |
| F3 | F8 |
| F4 | F9 |
| F5 | F10 |

2.8. Sélection des attributs vidéo ou de la couleur

Vous avez maintenant la possibilité de changer les couleurs de l'écran grâce à la nouvelle instruction **SET COLOR TO** sur l'IBM-PC, le VICTOR, le TEXAS-PC et tous les matériels compatibles IBM-PC.

La syntaxe est la suivante :

SET COLOR TO <n1,n2>

n1 représente la valeur à utiliser pour sélectionner la couleur des zones de saisie (**GET**), **n2** celle de la couleur pour l'affichage (**SAY**) ou lors de la frappe au clavier.

Valeurs par défaut :

| | | |
|-----------------|-------------------|-------------------------------|
| n1 = 112 | (inversion vidéo) | affichage noir sur fond blanc |
| n2 = 7 | (vidéo normale) | affichage blanc sur fond noir |

Couleurs possibles :

| Moniteur couleur | |
|------------------|-----------------|
| 0 | noir |
| 1 | bleu |
| 2 | vert |
| 3 | cyan |
| 4 | rouge |
| 5 | magenta |
| 6 | jaune |
| 7 | blanc |
| 112 | inversion vidéo |

Ex. : **SET COLOR TO 4,1**

Les zones de saisie (**GET**) seront en rouge sur fond noir, les zones (**SAY**) d'affichage seront en bleu.

III - dBASE II, Version 2.4

La version 2.4 est désormais disponible en français pour les systèmes d'exploitation CP/M-80, MP/M-80, CP/M-86 et MS-DOS.

Toutes les versions antérieures à la 2.4, acquises depuis le 1er septembre 1983 seront mises à jour gratuitement. Celles acquises à une date antérieure entreront dans le cadre des mises à jour payantes (1.350,00 Frs HT - logi-ciel + complément manuel).

1 - Nouvelles caractéristiques

- Tous les fichiers **OVERLAYS** et le fichier de messages constituent un seul et même fichier : **DBASEOVR.COM**.
- Les versions de CP/M-80 antérieures à 2.0 ne permettent pas l'utilisation de dBASE II version 2.4.
- Les fichiers mémoire (**.MEM**) générés par des versions antérieures à la 2.4 ne sont pas compatibles. Il sera nécessaire de les recréer.

2 - Nouvelles commandes et nouvelles fonctions de commande existantes

- Une commande **TEXT - ENDTEXT** vous permet maintenant d'afficher ou d'imprimer un paragraphe de texte sans utiliser : **?**, **??**, ou **SAY**.

Exemple : **? "à l'attention de" + NOM**

TEXT

Ceci est un exemple de la souplesse d'insertion d'un paragraphe en vue d'un affichage ou d'une impression grâce à la commande **TEXT**.

ENDTEXT

? "

Salutations"

- Il est maintenant possible d'utiliser un adressage relatif lors de l'utilisation des commandes **SAY** ou **GET**.

Exemple : **à 3,40 SAY DATE**
à \$+2,5 SAY "A l'attention de"
à \$,\$+2 SAY SEXE +" "+ NOM

- La commande **RELEASE** peut maintenant être sélective :

a) **RELEASE ALL LIKE <masque>**

efface toutes les variables mémoire correspondant au masque.

b) **RELEASE ALL EXCEPT <masque>**

efface toutes les variables mémoire sauf celles correspondant au masque.

Le **<masque>** peut utiliser le point d'interrogation pour masquer un caractère, un astérisque pour masquer tous les caractères existants.

Exemple : **RELEASE ALL LIKE B*** (efface toutes les variables mémoire dont la première lettre est un B)

- La commande **SAVE** peut maintenant être sélective.

SAVE ALL LIKE <masque> to <fichier mémoire>

sauvegarde toutes les variables mémoire correspondant au masque.

Exemple : **SAVE ALL LIKE ?DATE TO DATE2** (sauvegarde les variables ADATE, BDATE, ZDATE, etc...)

- Une commande d'aide : **HELP** vous permet d'obtenir immédiatement la description ainsi que la syntaxe d'utilisation des commandes de dBASE II. Tous les textes d'aide sont inclus dans le fichier DBASEMSG.TXT. Ce dernier pourra éventuellement être modifié par un traitement de texte ou la commande **MODIFY COMMAND** afin d'ajouter de nouvelles informations.

Exemple : **HELP CREATE**

- Une nouvelle fonction, **RANK**, retournera la valeur ASCII du premier caractère d'une chaîne.

Exemple : **. ? RANK("ABCD")**
. 65

- INDEX** (sans opérande) indexera l'enregistrement en cours. Très utile après la création d'un enregistrement par **APPEND BLANK**.
- La commande **CREATE** n'accepte plus deux noms de champs identiques.
- La commande **MODIFY STRUCTURE** vérifiera la syntaxe des différents champs.

- La commande **DISPLAY** peut désormais imprimer un ";" après l'exécution du **SET RAW ON**.
- ? et ?? ne placeront pas d'espaces entre les champs après l'exécution de **SET RAW ON**.
- La commande **REINDEX** mettra à jour le ou les fichiers index en cours d'utilisation sans qu'il soit nécessaire de les recréer.

Amélioration des possibilités de visualisation du répertoire du disque.

Exemple : **DISPLAY FILES LIKE FACT*.*** au lieu de **FACT????.***

- **EDIT, APPEND, CREATE** et **INSERT** peuvent utiliser un fichier de format (**SET FORMAT TO <nom de fichier>**). Les données saisies seront automatiquement placées dans la base de données en cours d'utilisation.

Exemple : **USE FACTURE**
SET FORMAT TO ECRFACT
APPEND
 (l'utilisateur complètera les zones vides afin de créer des enregistrements)

- La fonction **TYPE** renverra un "U" (undefined) si l'expression est mal orthographiée même dans le cas de positionnement correct des parenthèses.
- La commande **RESET** accepte un nom d'unité de disque.

Exemple : **RESET B**

- Tous les niveaux d'origine d'erreurs de syntaxe dans des fichiers de commandes seront spécifiés

Exemple : **ERREUR DE SYNTAXE**
ORIGINE DE L'ERREUR : FICHIER.CMD
***** DESIREZ-VOUS CORRIGER ET RECOMMENCER (Y/N) ?**

- Après un **PACK**, dBASE mettra à jour tous les fichiers index correspondants. Cette procédure est plus rapide qu'une compression du fichier de données et une mise à jour simultanée du ou des fichiers **INDEX**.
- **SET DELETE ON (OFF par défaut)** indique à dBASE d'ignorer tous les enregistrements repérés pour effacement. Ainsi les commandes **FIND** et **LIST** ne prendront en compte que les enregistrements non repérés. La seule exception est le positionnement sur un enregistrement repéré pour effacement en fonction de son numéro d'ordre.
- **DISPLAY STATUS** vous indiquera les noms des fichiers de données en cours d'utilisation, des fichiers index avec leur expression-clé, la date du jour, l'unité de disque active, ainsi que l'état de toutes les options modifiables par **SET**.

Exemple : **DISPLAY STATUS**

- La commande **UPDATE** effectuera la mise à jour en accès direct si le fichier en cours d'utilisation est indexé. (Le fichier d'origine aura une organisation quelconque). Le remplacement de données pourra s'effectuer entre des champs de noms différents (**REPLACE X WITH Y**).
- La commande **RESTORE** peut maintenant cumuler les variables d'un fichier mémoire avec celles en cours d'utilisation.

Exemple : **RESTORE FROM <fichier mémoire> ADDITIVE**

- Une nouvelle option de la commande **READ : NOUPDATE**, indique à dBASE II qu'il ne sera pas nécessaire de vérifier et de mettre à jour le fichier index en cours d'utilisation après chaque **READ**, dans le cas où aucun champ de clé ne risque d'être modifié. L'utilisation de cette option vous permettra de gagner du temps.
- La tentative d'ouverture d'un fichier de commandes déjà ouvert ou celle d'un fichier de données dans la base secondaire ouvert dans la base primaire entraînera le message d'erreur suivant :

*** FICHIER EN COURS D'UTILISATION

Ceci évitera le dépassement du nombre maximum (16) de fichiers ouverts, lors de l'exécution répétitive d'un même fichier de commandes ainsi que des confusions dans les fichiers de données.

3.1. Corrections de la Version 2.4

- Lors de l'utilisation de la commande **APPEND** sur une structure de plus de 23 champs, un retour chariot sur le premier champ du deuxième écran terminait la saisie.
- Les problèmes rencontrés lors de l'utilisation de la commande **INSERT** sans saisie de données sont résolus.
- Le problème de l'ajout (**APPEND**) ou de copie (**COPY**) de ligne nulle est résolu.
- L'utilisation de **APPEND** sur un fichier vide ne créera aucune information erronée en mode **SET CARRY ON**.
- Les problèmes rencontrés lors de l'utilisation de **BROWSE** sur des champs d'une longueur multiple de 80 caractères sont résolus.
- La commande **BROWSE** ne prend plus en compte **CTRL-T** et **CTRL-N**, ceux-ci entraînaient des erreurs dans les fichiers de données.
- Les problèmes de la fonction **TEST** sont résolus.
- La concaténation d'une chaîne de plus de 255 caractères entraînera un message d'erreur.
- La commande **GOTO RECORD <n>** positionne correctement le fichier index.
- Les manipulations internes d'empilement de **.AND** sont résolues.
- Les problèmes occasionnellement rencontrés avec **LOOP** (saut inconditionnel) sont résolus.
- Les problèmes d'utilisation de fichiers formats vides sont résolus.
- Les erreurs de manipulation et la commande **CANCEL** ferment maintenant tous les fichiers de commandes.
- **SET ALTERNATE TO** sans nom de fichier ne créera pas un fichier **.TXT** vide.
- Après un **RENAME**, dBASE "oubliera" l'ancien nom de fichier.
- Les problèmes de positionnement du pointeur de fin de fichier (EOF) après un **PACK** sur un fichier indexé vide sont résolus.
- Après un incident système, **PACK** permet maintenant de récupérer des enregistrements en se référant à la position de la fin de fichier.
- Les problèmes de saisie de données numériques en mode Plein écran avec **SET CONFIRM ON** sont résolus.

- . Amélioration des manipulations numériques en mode Modification plein écran
- . **SET LINKAGE ON** permet maintenant de traiter un ensemble de fichiers indexé et non indexés
- . **SET ESCAPE ON** est maintenant pris en compte avec les commandes **ACCEPT** et **INPUT**.
- . **DISPLAY STRUCTURE** d'une structure vierge ne pose plus aucun problème.
- . **DISPLAY FILES LIKE *.*** n'interfère plus avec un fichier défini par **SET ALTERNATE**
- . Le cumul de la taille des variables est correctement effectué avec **DISPLAY MEMORY**
- . Le problème rencontré lors de l'utilisation de **TOTAL FOR <condition>**, où cette <condition> n'est jamais vraie, est résolu
- . **SUM TO <variable>** permet maintenant à la <variable> d'avoir une longueur de 10 caractères
- . Les problèmes de la commande **SAY** de valeurs numériques sont résolus.
- . La précision **FIELDS** de la commande **TOTAL** ne détermine pas les champs du fichier de destination
- . L'abandon de la création d'un fichier de report ne créera plus un fichier **.FRM** vide
- . Les problèmes rencontrés lors de l'apparition d'une erreur de syntaxe dans la définition d'une expression de sous-total d'un fichier de report sont maintenant résolus
- . La commande **REPORT** permet maintenant d'utiliser 24 champs pour les totaux et les sous-totaux
- . Les erreurs de syntaxe et la commande **CANCEL** terminent un éventuel **DO CASE**
- . Les différents cas dans un **DO CASE** peuvent maintenant être indentés (présentation en escalier) sans problème
- . Amélioration de l'éditeur de texte **MODIFY COMMAND**
- . **MODIFY COMMAND** ne créera plus un fichier **AD.CMD** si vous frappez **MODIFY COMMAD**
- . Les perturbations rencontrées dans les fichiers de données à l'issue d'un tri (**SORT**) sont maintenant résolues

IV - PROBLÈMES ET SOLUTIONS

- 4.1. Pour quelle raison, après la saisie de 150 enregistrements, seuls les 32 premiers sont accessibles (LIST), bien que la structure indique l'existence de 150 enregistrements ?

Ceci est dû à un mauvais positionnement du pointeur de fin de fichier (EOF - End of File). La seule possibilité de mauvais positionnement du pointeur EOF est consécutive à une sortie de dBASE par une autre commande que QUIT (reset système, coupure secteur, etc.) lorsque les fichiers de données ne sont pas fermés.

Vous trouverez ci-dessous les commandes à exécuter afin de recouvrer la totalité des enregistrements d'un fichier dans lequel se trouve un code de fin de fichier entre deux enregistrements.

Effectuer une copie de sauvegarde avant d'exécuter les commandes suivantes :

1. USE <nom de fichier altéré>
2. COPY TO <nouveau fichier>
3. GO &+2
4. COPY NEXT 65535 TO <TEMP>
5. USE <nouveau fichier>
6. APPEND FROM <TEMP>
7. DELETE FILE <TEMP>
8. DELETE FILE <nom du fichier altéré>
9. CLEAR
10. RENAME <nouveau fichier> TO <nom du fichier altéré>

Dans le cas où il existerait plus d'un code de fin de fichier (1AH) mal positionnés, vous pourrez réitérer les lignes 3 et 4 afin de vous positionner sur tous les pointeurs de fin de fichier mal positionnés.

- 4.2. Comment écrire des fichiers de commande avec un autre éditeur de texte que MODIFY COMMAND sans perdre de lignes à la fin de celui-ci ?

La version 2.4 de dBASE II pour les matériels 16 bits nécessite que le pointeur de fin d'un fichier de commande soit positionné à la fin d'un bloc de 512 octets. Certains éditeurs de texte ou traitements de texte positionnent le pointeur de fin de fichier après le dernier caractère du fichier de commande et non pas après un bloc d'une taille physique de 512 octets. Lorsque vous exécutez un fichier de commande dont la taille physique n'est pas multiple de 512 octets, les lignes après le dernier bloc de 512 octets ne sont pas lues.

Pour solutionner ces problèmes, vous pouvez utiliser l'une des trois possibilités suivantes :

1. - Exécuter **MODIFY COMMAND** sur votre fichier de commande
- Se Positionner à la fin de celui-ci
- Ajouter les éventuelles lignes manquantes
- Frapper **CTRL-W** pour sauvegarder.
2. Ajouter deux lignes de 64 astérisques ou plus à la fin de votre fichier de commande avant de le sauvegarder depuis votre traitement de texte habituel. (Apparemment, 128 octets dans le bloc de 512 sont suffisants pour permettre sa lecture).
3. Placer la disquette-système dans l'unité A, la disquette contenant le fichier de commande dans l'unité B. Frapper les commandes suivantes :

DEBUG <nom de fichier>.PRG

RCX (CR)

CX 0153

:200

W (CR)

Q (CR)

remplacer le contenu du registre CX par la valeur multiple de 512 directement supérieur à son contenu

- 4.3. Une application fonctionnant sans erreur avec la version 2.3 affiche le message : FICHIER EN COURS D'UTILISATION avec la version 2.4

Ceci est dû à une nouvelle possibilité de la version 2.4 qui vous informe que vous essayez d'exécuter un fichier de commande déjà ouvert précédemment et non refermé, ou que vous utilisez un même fichier de données dans les zones primaire et secondaire. La fermeture d'un fichier de commande est obtenue par l'exécution de l'instruction **RETURN**. Avec la version 2.3, il vous est possible d'ouvrir plusieurs fois le même fichier de commande à concurrence de 16 fichiers (limite du nombre de fichiers ouverts au même instant). Ceci entraînait l'apparition du message **TROP DE FICHIERS SONT OUVERTS** dès que le maximum de 16 fichiers ouverts était atteint, ce qui laissait dans l'expectative l'utilisateur qui ne pouvait expliquer l'origine de cette erreur. Il est donc préférable que cette erreur de programmation soit signalée immédiatement. Nous vous rappelons que le retour au fichier de commande appelant (fichier de commande précédent) est obtenu par l'instruction **RETURN** qui fermera également le fichier de commande qui la contient.

- 4.4. Pourquoi le curseur se positionne-t-il de façon incorrecte après plusieurs saisies plein écran avec l'instruction **GET** ?

Le nombre maximum d'instructions **GET** entre deux effacements d'écran ou l'utilisation de la commande **CLEAR GETS** est de 64. Si vous excédez cette limite, vous serez alors confrontés à des erreurs de positionnement. Si seul le contenu des zones de saisie de votre écran est modifié, vous devez utiliser : **CLEAR GETS** après l'instruction **READ** afin de réinitialiser tous les **GET** précédemment émis.

- 4.5. Lors de l'accès à un enregistrement se trouvant avant le dernier enregistrement du fichier, apparition du message d'erreur "ENREGISTREMENT NON SITUE"

Ceci peut se produire :

- Si vous utilisez un fichier index altéré ou dans le cas de pointeur de fin de fichier mal positionné.

Si vous utilisez un fichier **INDEX**, exécutez la commande **REINDEX**. Si l'erreur n'a pas disparu, utilisez la procédure précédente (4-1) afin de négliger le pointeur de fin de fichier mal positionné.

- Si l'en-tête du fichier de données est altéré, exécutez alors la procédure suivante :

```
USE <fichier altéré>
COPY STRUCTURE TO <nouveau fichier>
USE <nouveau fichier>
APPEND FROM <fichier altéré>
DELETE FILE <fichier altéré>
USE
RENAME <nouveau fichier> TO <fichier altéré>
```

- 4.6. Instruction **FILE(<nom de fichier>)**

Si vous utilisez l'instruction **FILE(<nom de fichier>)** sur un fichier en cours d'utilisation, cela entraînera une disparition de fichiers du répertoire du disque.

Fermez les fichiers en cours (**USE (CR)**) avant de vérifier l'existence d'un fichier de données.

- 4.7. **JOIN**

Lors de l'exécution de la commande suivante :

```
JOIN TO <nouveau fichier> FOR <champ> = S.<champ>
```

créera des noms de fichiers en double dans la structure du <nouveau fichier>.

Il suffit de créer la structure du **<nouveau fichier>** avant l'exécution de la commande **JOIN**. Cette commande ne crée la structure que si le fichier n'existe pas.

4.8. MODIFY COMMAND

Lors de la modification d'un fichier de commande, il arrive parfois que dBASE charge le fichier de commande de sauvegarde (.BAK) au lieu du fichier de commande exécutable (.CMD ou .PRG).

Afin d'éviter ce genre d'erreur, il est nécessaire de préciser l'extension .CMD (version 8 bits) ou .PRG (version 16 bits).

Ex. : **MODIFY COMMAND MENU.PRG (CR)**

4.9. Utilisation de l'instruction USING

L'instruction **USING** ne permet pas d'obtenir le résultat escompté. Par exemple :

```
STORE "123456" TO CHAINE
à <ligne>,<colonne> SAY CHAINE USING "99/99/99"
```

affichera : "12/45/ "
au lieu de : "12/34/56"

Il est nécessaire d'utiliser la fonction de sous-chaîne de caractères. Par exemple :

```
STORE "123456" TO CHAINE
à <ligne>,<colonne> SAY $(CHAINE,1,2)+" / "+
                        $(CHAINE,3,2)+" / "+
                        $(CHAINE,5,2)
```

affichera : "12/34/56"

4.10. Saisie de caractères codés sur 8 bits (caractères accentués sur l'IBM-PC par exemple) avec l'éditeur de texte MODIFY COMMAND sur la version 2.4 MS-DOS de dBASE II

Lors de la création d'un fichier de commande avec **MODIFY COMMAND**, vous avez la possibilité de saisir des caractères codés sur 8 bits et de les sauvegarder. Par contre, en cas de modification, le chargement du fichier de commande s'effectue sur 7 bits ce qui entraîne la transcription de tous les caractères codés sur 8 bits (81H 01H), en outre les caractères accentués des claviers de certains micro-ordinateurs 16 bits (IBM-PC).

Vous trouverez ci-dessous la modification à effectuer pour permettre un chargement sur 8 bits.

Placez votre disquette-système dans l'unité A et la disquette dBASE II dans l'unité B.

```
A>DEBUG B:DBASEOVR.COM (CR)
- E6 845 7F FF (CR)          (remplacer 7FH par FFH)
- W (CR)
- Q
A>
```

V - UN SUPPORT TECHNIQUE TÉLÉPHONIQUE A VOTRE DISPOSITION

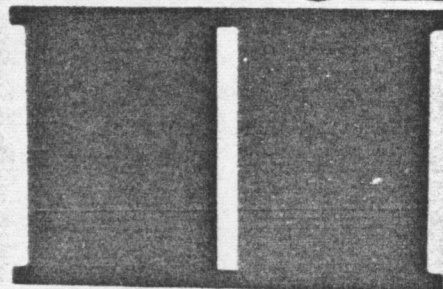
Dans le cas de problèmes techniques non référencés dans la documentation qui vous a été fournie, vous pouvez obtenir tout renseignement complémentaire auprès de notre service de support téléphonique.

Du lundi au jeudi : de 8h30 à 12h30 et de 13h30 à 17h30
Le vendredi : de 8h30 à 12h30 et de 13h30 à 16h30

Pour bénéficier de ce service, il vous faudra fournir :

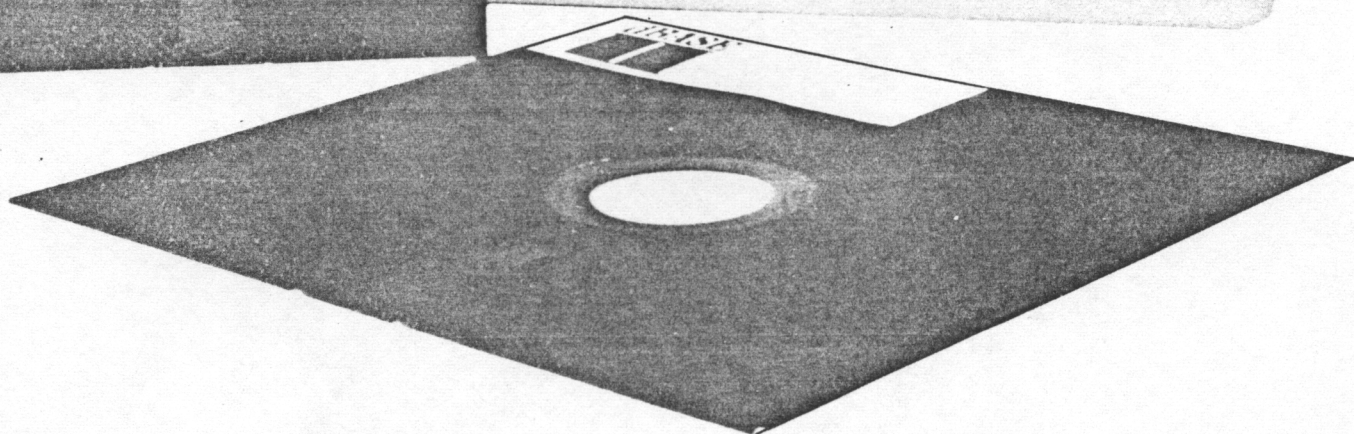
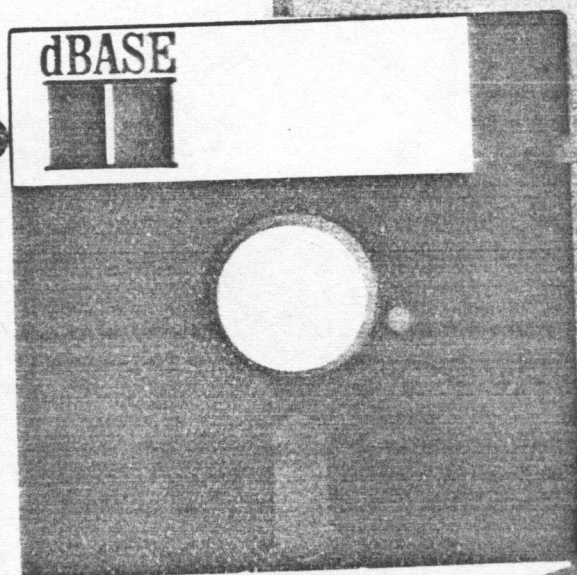
- votre numéro de licence d'exploitation
- vos coordonnées
- la version de dBASE II
- le système d'exploitation
- le type de votre micro-ordinateur.

dBASE



Système de gestion de base
de données relationnelle en
langage machine

Ashton-Tate



dBASE II UN PRODIGIEUX OUTIL DE PROGRAMMATION

dBASE II est un langage puissant permettant de réaliser complètement tout programme de gestion sous une forme aussi perfectionnée que ceux des grands systèmes informatiques.

dBASE II vous permet de créer :

- les grilles de saisie
- les menus
- les fichiers relationnels
- les tris et sélections les plus complexes
- les rapports d'impression à votre goût

dBASE II est disponible avec messages, manuel et guide de formation en français sous CP/M 2.x, CP/M 86, MS-DOS, VISIOTEXTE, IBM et IBM PC.

Les détenteurs de version américaine peuvent acquérir une mise à jour à la nouvelle version 2.3B en français qui comporte maintenant le créateur de grille de saisie ZIP.

Le système de gestion de bases de données, le plus puissant et le plus simple, pour micro-ordinateur

dBASE II traite la gestion de bases de données.

C'est un programme de gestion d'informations, et non un programme de gestion de fichiers.

Il est relationnel, et non hiérarchique ; ainsi, n'avez-vous pas à vous occuper des ensembles, des listes, des pointeurs et toutes les autres complexités des anciens systèmes.

Il est interactif, la saisie des données est facile.

Il peut être programmé de telle sorte que même du personnel non spécialisé peut exécuter avec précision vos applications, même les plus compliquées.

Il vous donne un plein contrôle X-Y de telle sorte que l'entrée et la sortie sont faites comme vous désirez qu'elles le soient, même sur vos imprimés existants.

Il a son propre langage. Ainsi est-il facile à utiliser, avec des structures qui facilitent sa bonne utilisation.

Il suffit de quelques commandes simples pour changer, étendre, modifier des bases de données.

La documentation est si complète qu'elle contient une véritable autoformation de gestion de bases de données (voir la table des matières partielle).

Que pouvez-vous faire avec ?

Avec dBASE II, vous pouvez étendre la puissance de votre micro-ordinateur à des travaux qui étaient auparavant réservés aux plus gros ordinateurs.

Voici une liste partielle des applications pour lesquelles dBASE II a été utilisé :

- Grand Livre
- Journal des Comptes
- Comptabilité Clients

```
CREATE
DONNEZ LE NOM DU FICHIER : FACTURE
DONNEZ LA STRUCTURE SUIVANT LE FORMAT:
CHAMP  NOM,TYPE,NB CARACTERES,NB DECIMALES
001    CTRL:DATE,C,7
002    CTRL:NMNR,C,7
003    CLIENT,C,3
004    LOT:NMNR,N,3
005    NOM,C,20
006    DESIGNATION,C,15
007    PRIX,N,9,2
008    FACT:DATE,C,7
009    FACT:NMNR,C,7
010    HEURE,N,6,2
011
SAISIE DES DONNEES MAINTENANT? N
```

- Comptabilité Fournisseurs
- TVA
- Feuille de Paie
- Gestion et Ecriture des Chèques
- Facturation
- Gestion de Stock
- Comptabilisation des Travaux
- Calcul de l'Impôt
- Renvois de Documents
- Comptabilité générale, plan comptable
- Planification
- Etiquettes d'Expédition
- Evénements calendaires

Si votre application nécessite la gestion de données, dBASE II peut être la réponse.

Il est facile à utiliser

Vous pouvez créer une base de données et commencer à y entrer des données en moins d'une minute.

Tapez CREATE, puis répondez aux messages-guides vous demandant d'affecter un nom au fichier et de définir les zones de vos enregistrements, comme représenté sur la première photo.

Une fois l'enregistrement défini, vous pouvez commencer à entrer immédiatement des données, ou ajouter ultérieurement des informations en tapant APPEND. Dans les deux cas, dBASE II vous offre une structure d'enregistrement complète pour laquelle vous remplissez simplement une partie ou la totalité des blancs.

```
*****FICHIER DE COMMANDES POSTFACT*****
Entrez les factures dans POSTAGE comme elles se présentent. Un fichier
provisoire appelé FACT:EN est utilisé de sorte que seule la structure
nécessaire est affichée sur l'écran pour la saisie. Après la saisie, les
enregistrements sont transférés dans POSTAGE et FACT:EN est supprimé. Le
fichier de commandes COUTEMPO est utilisé plus tard pour transférer ces
dernières dans COUTDEFI une fois qu'elles ont été vérifiées.
*****
ERASE
USE POSTAGE
COPY TO FACT:EN STRUCTURE FIELDS Client, Travail:Nmro,;
Nom, Descrip, Facture:Date, facture:Nmro, Montant
USE FACT:EN
STORE 'O' TO Continue
DO WHILE ! (Continue) = 'O'
    APPEND
    ? 'Est-ce qu'il y a encore des FACTURES FOURNISSEURS à
    ? 'saisir (O ou N) ?
    WAIT VARS Continue
ENDDO
USE POSTAGE
APPEND FROM FACT:EN
ERASE
DELETE FILE FACT:EN
RETURN
```

Pour créer une base de données, il suffit de lui donner un nom, puis de décider de la taille des zones et du type de données que vous allez y stocker.

Vous pouvez déplacer, copier ou manipuler de différentes façons des bases de données sans avoir à réintroduire vos données.

Transformation des données en information

Vous pouvez travailler de façon interactive (en mode conversationnel) avec votre base de données ou avec des programmes que vous créez en utilisant les commandes dBASEII.

La plupart des commandes dBASEII sont des verbes, auxquels sont ajoutées des structures de programmation qui facilitent l'assemblage de programmes clairs, même si vous n'êtes pas programmeur.

Si vous désirez répéter n'importe quelle action que vous avez faite, réunissez les commandes dans un fichier de commandes, puis n'importe qui dans votre bureau peut exécuter la série d'opérations avec une seule commande (DO PAIE, DO POSTAGE, DO FICHIER de commande).

Dans tous les cas, vous avez le contrôle de toutes les opérations sur les bases de données.

Afficher une partie ou la totalité des données pour n'importe quelle condition que vous désirez appliquer. Manipuler les données à partir de bases de données entières jusqu'à de simples caractères dans une zone de données. Changer vos données ou la structure entière de votre base de données avec quelques commandes, sans réintroduire la totalité de la base de données. Faire des calculs automatiques sur des zones, des enregistrements, et des bases de données entières avec quelques manipulations.

Pour un traitement des données encore plus rapide, INDEXez vos bases de données. Avec cette fonction, vous pouvez retrouver n'importe quel enregistrement en moins de deux secondes, pour une base de données de taille moyenne à grande taille, même en utilisant des disques souples. Aucun autre système n'approche cette performance.

Pour la sortie des informations, dBASEII offre de nombreuses possibilités. La commande REPORT parcourt vos bases de données, et sélectionne toutes les informations que vous désirez, puis les totalise, les sous-totalise et les présente exactement comme vous le désirez.

Lorsque vous imprimez les informations que dBASEII a calculées, vous pouvez placer n'importe quel caractère ou n'importe quelle valeur exactement où vous le désirez sur la page en utilisant une simple commande, car il vous donne un plein contrôle X-Y. Contrairement aux autres systèmes, dBASEII ne vous oblige pas à changer votre façon de travailler et vous permet d'utiliser vos imprimés existants.

REPORT FORM lotfab TO PRINT FOR LOT:NER-750

PAGE NO. 00001
19/11/82

| DATE | FOURNISSEURS | DESIGNATION | PRIX |
|-------------|----------------|---------------|---------|
| 800615 | PHOTOMATIC | cordon decl. | 65,20 |
| 800821 | PHOTO BOUTIQUE | objectif zoom | 25,00 |
| 800521 | ALPHA | etui | 150,10 |
| 800620 | MEGADIAPO | pied rotule | 320,50 |
| 810224 | VISIONCLUB | flash | 420,00 |
| 810312 | INTERPHOTO | tele objectif | 1250,50 |
| 810524 | WONDAIRE | piles | 15,00 |
| ** TOTAL ** | | | 2246,30 |

Des données qu'il faudrait des mois à organiser peuvent l'être en quelques minutes avec une simple ligne de commande.

Vous ne pourrez plus
vous en passer



TABLE DES MATIERES

1ère Partie

Introduction
Configuration nécessaire
Spécifications dBASEII
Principes de Base des bases de données
Enregistrements, Fichiers et Types de Données dBASEII

2ème Partie

Préparation
Comment créer (CREATE) un fichier
Travail avec des bases de données multiples (USE, SELECT, CLEAR)
Vérification de la structure
Modification de la structure de votre base de données (MODIFY)
Entrée des données (APPEND)
Sélection des enregistrements et des données (LIST, DISPLAY, ?)
Modification des données et des enregistrements (EDIT, INSERT, DELETE, RECALL, PACK)

Fonctions d'Édition Plein Écran

3ème Partie

Utilisation des expressions pour la sélection et le contrôle
(champs, variables mémoire, constantes, opérateurs)
Opérations dBASEII
Extension de votre contrôle avec des fonctions.
Modification rapide des données (CHANGE, REPLACE)
Fusion d'enregistrements et de fichiers (APPEND, UPDATE)
Fusion de bases de données. (JOIN)
Génération de fichiers (en utilisant des données existantes).
Changement de nom et restructuration de la base de données (COPY, MODIFY, APPEND, RENAME)
Organisation de vos bases de données (SORT, INDEX).
Commandes de positionnement (SKIP, GO ou GOTO, LOCATE, FIND)
Commandes système.
Changement des caractéristiques dBASEII (SET)

4ème Partie

Sortie formatée des informations (REPORT)
Comment automatiser vos traitements d'informations (fichiers de commandes CMD, DO, RETURN)
Quelques mots sur la programmation et sur la planification de vos fichiers de commandes.
Structure des commandes dBASEII (IF..ELSE..ENDIF, DO WHILE, LOOP, ENDDO, CANCEL)
Travail avec des variables mémoire (STORE et RELEASE, SAVE et RESTORE)
Comptage et totalisation automatiques (COUNT, SUM, TOTAL)
Entrée des données de façon interactive pendant une exécution (WAIT, INPUT, ACCEPT)

dBASEII est accompagné d'une documentation très complète en français, vous permettant d'utiliser facilement ses possibilités.

Ashton-Tate

COMMANDES dBASE II

CREATION DE FICHIERS

CREATE crée de nouvelles bases de données structurées.

COPY copie des bases de données, leur structure, ou leurs données dans un nouveau fichier.

REPORT génère des informations selon vos spécifications, sélectionnant uniquement les informations que vous désirez, avec ou sans les totaux et les sous-totaux.

SAVE copie des variables mémoire dans un fichier pour une utilisation ultérieure.

INDEX crée un fichier d'index pour une localisation plus rapide des données.

AJOUT DES DONNEES

APPEND ajoute des enregistrements à la fin d'une base des données.

CREATE permet l'entrée de données lorsque le fichier est créé.

INSER insère des enregistrements dans un fichier.

EDITION DE DONNEES

EDIT modifie des enregistrements et des zones spécifiques de bases de données.

REPLACE change le contenu de zones spécifiées.

CHANGE édite des zones spécifiées de la base de données.

DELETE positionne l'indicateur de suppression.

RECALL efface l'indicateur de suppression.

PACK élimine les enregistrements à supprimer.

AFFICHAGE DE DONNEES

DISPLAY et **LIST** affichent des enregistrements, des zones et des expressions.

? affiche la valeur de variables ou d'expressions.

REPORT met en forme et affiche des données selon les spécifications, avec ou sans les totaux.

READ affiche les informations et les données.

SUM totalise les zones que vous spécifiez dans une base de données.

TOTAL totalise les zones spécifiées dans une nouvelle base de données.

MANIPULATION DE FICHIERS

DO lance l'exécution d'une séquence de fichiers de commandes.

APPEND ajoute des données provenant d'autres fichiers (même non dBASE II).

SELECT permet de basculer entre deux bases de données utilisées.

SORT organise la base de données triée sur une zone.

COMMANDE DE VARIABLES MEMOIRE

ACCEPT stocke des données alphabétiques pour une utilisation ultérieure.

INPUT accepte des données alphabétiques, numériques ou logiques.

WAIT accepte une entrée d'un seul caractère.

GET fonctionne comme la commande **INPUT**, utilisé dans le mode d'entrée interactif plein écran.

SUM sauvegarde des totaux dans des variables mémoire.

SAVE stocke des variables mémoire dans un fichier disque.

RESTORE retrouve les variables mémoires stockées.

AUTRES COMMANDES

FIND localise un enregistrement indexé, en moins de deux secondes.

LOCATE trouve les enregistrements qui répondent aux conditions spécifiées.

SKIP vous permet de vous déplacer en avant ou en arrière dans la base de données.

DO WHILE permet des opérations répétitives.

IF... ELSE vous permet de faire des choix.

CONFIGURATION NECESSAIRE

dBASE II nécessite les environnements matériel et logiciel suivants :

- Système commandé par microprocesseur 8080, 8085 ou Z-80 (comme le Northstar, ITT 3030, Zenith Z100, Superbrain, Xerox 820, Altos, Apple II avec la carte Z-80, etc...).
- 48K octets minimum de mémoire (dBASE II utilise des emplacements de 5CH à A400H).
- 128 K octets minimum sous MSDOS ou CP/M 86.
- Le système d'exploitation CP/M (version 2.x), MP/M.
- Le système d'exploitation MS-DOS ou CP/M 86 (pour la version 16 bits).
- Un écran de visualisation avec curseur adressable, 24 lignes par 80 colonnes, si des opérations plein écran doivent être utilisées.
- Une imprimante texte optionnelle (pour certaines commandes).
- Disponible en formats Visiotexte IBM et IBM PC.

CARACTERISTIQUES dBASE II

| | |
|--|------------------------------------|
| Enregistrements par fichier | |
| de base de données | 65535 max |
| Caractères par enregistrement | 1000 max |
| Zones par enregistrement | 32 max |
| Caractères par zones | 254 max |
| Plus grand nombre | + ou - $1,8 \times 10^{63}$ approx |
| Plus petit nombre | + ou - 1×10^{-63} approx |
| Précision numérique | 10 chiffres |
| Longueur des chaînes de caractères ... | 254 caractères max |
| Longueur des lignes de commandes .. | 254 caractères max |
| Longueur des en-têtes d'état | 254 caractères max |
| Longueur des clés d'index | 100 caractères max |
| Expression dans la commande SUM | 5 max |

dBASE II est une marque déposée d'ASHTON-TATE

MANUEL EN FRANÇAIS



La Commande Electronique

5 VILLA DES ENTREPRENEURS - 75015 PARIS

TEL (1) 577.31.82

TELEX 204 237 F