بررسی انواع فشرده سازی در پایگاه داده ها

علی باقریان و جواد رضاپوریان

۱- دانشجوی کارشناسی نرم افزار کامپیوتر دانشکده فنی پسران شهرکرد Email:alibagherian13821382@gmail.com
۲- مهندس دانشکده فنی پسران شهرکرد فنی و حرفه ای پسران شهرکرد Email: Javad3159@yahoo.com

اطلاعات مقاله

تاریخچه مقاله:
تاریخ دریافت مقاله: ۱۴۰۲/۱۰/۰۲
تاریخ پذیرش مقاله: ۱۴۰۲/۱۱/۰۴
تاریخ انتشار مقاله: ۱۴۰۲/۱۱/۰۶

کلمات کلیدی:
فشرده سازی
پایگاه داده
سیستم مدیریت محتوا
داده ها
خلاصه سازی
SQL

فشرده‌سازی در پایگاه‌های داده از روش‌های حیاتی برای بهینه‌سازی فضا و افزایش کارایی استفاده از منابع ذخیره‌سازی است. این فرایند از طریق کاهش حجم داده‌ها به وسیله الگوریتم‌ها و روش‌های مختلفی انجام می‌شود. با توجه به این که فضای ذخیره‌سازی و پردازش در پایگاه‌های داده می‌تواند هزینه‌های قابل توجهی داشته باشد، استفاده از فشرده‌سازی می‌تواند به طور چشمگیری کارایی را افزایش دهد. این روش می‌تواند از طریق کاهش حجم داده‌ها، زمان لود و ذخیره‌سازی را بهبود بخشیده و در نتیجه عملکرد کلی سیستم پایگاه داده را بهبود بخشد. همچنین، فشرده‌سازی می‌تواند امنیت را نیز تقویت کرده و موجب کاهش نیاز به فضای ذخیره‌سازی و پهنای باند شبکه شود. با اینکه فشرده‌سازی ممکن است برخی محدودیت‌ها و هزینه‌های محاسباتی را به همراه داشته باشد، اما مزایای آن از نظر بهینه‌سازی منابع و افزایش کارایی، بسیار چشمگیر است.

۱ – مقدمه و هدف

در دهه‌های اخیر، حجم عظیمی از داده‌ها و اطلاعات به سرعت در پایگاه‌های داده جمع‌آوری و ذخیره می‌شوند. این افزایش چشمگیر حجم داده‌ها، نیازمند راهکارهایی برای بهینه‌سازی استفاده از فضای ذخیره‌سازی و بهبود عملکرد سیستم‌های پایگاه داده می‌شود. یکی از راهکارهای کلیدی برای این منظور، استفاده از تکنیک‌های فشرده‌سازی در پایگاه‌های داده است. فشرده‌سازی به عنوان یک روش موثر برای کاهش حجم داده‌ها و بهبود عملکرد در جستجوها، ذخیره‌سازی و انتقال داده‌ها شناخته می‌شود.(۱)

در این مقاله، ما به بررسی مفاهیم و انواع مختلف فشرده‌سازی در پایگاه‌های داده می‌پردازیم. هدف اصلی ما، بررسی تأثیرات مثبت و منفی استفاده از فشرده‌سازی بر عملکرد و کارایی سیستم‌های پایگاه داده است. همچنین، ما به بررسی مزایا و محدودیت‌های مختلف این روش می‌پردازیم و نقش آن را در بهینه‌سازی مصرف منابع و افزایش کارایی سیستم‌های پایگاه داده بررسی می‌کنیم. در این راستا، از منابع و مطالعات پیشین در زمینه فشرده‌سازی در پایگاه داده استفاده می‌کنیم و به دقت تأثیرات این تکنیک را بر کارایی سیستم‌های پایگاه داده بررسی می‌کنیم.(۲)

انواع فشرده سازی ها عبارتند از :

۱- فشرده سازی بر اساس سطح

۲- فشرده سازی بر اساس نوع داده

۳- فشرده سازی بر اساس مکان.

۱.فشرده سازی بر اساس سطح فشرده سازی

سطوح مختلف فشرده‌سازی در پایگاه‌های داده می‌توانند به عنوان یکی از مهم‌ترین عوامل موثر بر کارایی و میزان بهره‌وری سیستم‌های ذخیره و بازیابی داده مطرح شوند. فشرده‌سازی در سه سطح اصلی اعمال می‌شود: سطح سیستم فایل، سطح صفحات و سطح سطرها یا ستون‌ها.در سطح سیستم فایل، فشرده‌سازی اطلاعات در سطح کلی فایل‌های داده انجام می‌شود. این فشرده‌سازی معمولاً به صورت فشرده‌سازی داده‌های غیر فشرده یا استفاده از الگوریتم‌های فشرده‌سازی معمولی مانند gzip یا bzip2 انجام می‌شود که می‌تواند منجر به کاهش قابل توجه حجم داده و فضای مورد استفاده شده در دیسک شود.در سطح صفحات، فشرده‌سازی اطلاعات بر روی صفحات یا بلاک‌های داده‌ای صورت می‌گیرد. این فشرده‌سازی می‌تواند منجر به کاهش تعداد دسترسی‌ها به دیسک و افزایش سرعت بازیابی داده شود.در سطح سطرها یا ستون‌ها، فشرده‌سازی به صورت مستقیم بر روی ستون‌ها یا سطرهای داده‌ها اعمال می‌شود. این نوع فشرده‌سازی معمولاً برای داده‌های ساختار یافته مانند جداول رابطه‌ای استفاده می‌شود و می‌تواند به طور موثری حجم داده‌ها را کاهش داده و عملکرد عملیات جستجو و فیلترینگ را بهبود بخشد.با استفاده از سطوح مختلف فشرده‌سازی، می‌توان عملکرد سیستم‌های پایگاه داده را بهبود داده و مصرف منابع ذخیره‌سازی و پردازش را بهینه‌سازی کرد. (۲)

۱. فشرده سازی صفر یا بدون فشرده سازی (No compression):در این روش، هیچ گونه فشرده سازی انجام نمی شود و داده ها به صورت اولیه ذخیره می شوند. این روش ساده ترین و سریع ترین روش فشرده سازی است، اما فضای بیشتری را اشغال می کند.

۲. فشرده سازی ساده (Basic compression): این روش فشرده سازی ساده ای است که بر اساس حذف تکرارها و استفاده از کدهای کوتاهتر برای مقادیر تکراری انجام می شود. این روش نسبت به فشرده سازی بدون فشرده سازی فضای کمتری را اشغال می کند، اما همچنان نسبت به روش های پیشرفته تر فضای بیشتری را اشغال می کند.

۳. فشرده سازی پیشرفته (Advanced compression): این روش فشرده سازی پیشرفته تری است که بر اساس الگوریتم های پیچیده تر انجام می شود. این روش می تواند فضای بسیار کمتری را نسبت به روش های ساده تر اشغال کند، اما سرعت آن نیز کندتر است.

۱،۱فشرده سازی صفر یا بدون فشرده سازی(No compression):

فشرده سازی صفر یا بدون فشرده سازی ساده ترین و سریع ترین روش فشرده سازی است. این روش فضای زیادی را اشغال می کند، اما سرعت آن نیز بالا است.

فشرده سازی صفر یا بدون فشرده سازی ساده ترین و سریع ترین روش فشرده سازی است. این روش هیچ گونه تغییری در داده ها ایجاد نمی کند و داده ها به صورت اولیه ذخیره می شوند.

فشرده سازی صفر یا بدون فشرده سازی مزایای زیر را دارد:

- ساده و سریع هستند
- نیاز به پیاده سازی یا مدیریت ندارد

فشرده سازی صفر یا بدون فشرده سازی معایب زیر را نیز دارد:

- فضای ذخیره سازی بیشتری نسبت به روش های فشرده سازی پیشرفته اشغال می کند
- سرعت دسترسی به داده ها کمتر است

در مواردی که داده ها تکراری نیستند یا حجم داده ها کم است، استفاده از فشرده سازی صفر یا بدون فشرده سازی می تواند گزینه مناسبی باشد.

فشرده سازی صفر یا بدون فشرده سازی معمولاً برای پایگاه داده های کوچک یا پایگاه داده هایی که به داده ها دسترسی زیادی وجود ندارد، استفاده می شود.(۱)

مثال ۱:

فرض کنید یک جدول در پایگاه داده وجود دارد که حاوی ۱۰۰۰ رکورد است. هر رکورد شامل یک فیلد کاراکتر با طول ۱۰ کاراکتر است. در این حالت، حجم داده های جدول برابر با ۱۰۰۰۰ کاراکتر است.

اگر از فشرده سازی صفر یا بدون فشرده سازی استفاده شود، داده های جدول به صورت اولیه ذخیره می شوند. در این حالت، حجم داده های جدول همچنان برابر با ۱۰۰۰۰ کاراکتر است.

مثال ۲:

فرض کنید یک جدول در پایگاه داده وجود دارد که حاوی ۱۰۰۰۰ رکورد است. هر رکورد شامل یک فیلد عددی با طول ۸ بایت است. در این حالت، حجم داده های جدول برابر با ۸۰۰۰۰ بایت است.

اگر از فشرده سازی صفر یا بدون فشرده سازی استفاده شود، داده های جدول به صورت اولیه ذخیره می شوند. در این حالت، حجم داده های جدول همچنان برابر با ۸۰۰۰۰ بایت است.

۱,۲. فشرده سازی ساده (Basic compression):

فشرده سازی ساده روشی ساده برای فشرده سازی داده ها است که بر اساس حذف تکرارها و استفاده از کدهای کوتاهتر برای مقادیر تکراری انجام می شود. این روش معمولاً برای داده هایی که دارای تکرارهای زیادی هستند، مانند متن و رشته ها، استفاده می شود.

در فشرده سازی ساده، ابتدا داده ها بررسی می شوند تا مقادیر تکراری پیدا شوند. سپس، این مقادیر تکراری با یک کد کوتاهتر جایگزین می شوند. به عنوان مثال، اگر یک جدول حاوی چندین رکورد باشد که همه آنها دارای مقدار Address یکسان هستند، می توان این مقدار را با یک کد کوتاهتر فشرده کرد.

فشرده سازی ساده می تواند فضای ذخیره سازی را تا حد زیادی کاهش دهد. با این حال، این روش ممکن است باعث کاهش سرعت دسترسی به داده ها شود.

مزایا

- کاهش حجم داده های ذخیره شده
- افزایش سرعت انتقال داده ها

معایب

- ممکن است باعث کاهش سرعت دسترسی به داده ها شود

فشرده سازی ساده روشی ساده و موثر برای فشرده سازی داده ها است. این روش می تواند فضای ذخیره سازی را تا حد زیادی کاهش دهد و سرعت انتقال داده ها را افزایش دهد. با این حال، این روش ممکن است باعث کاهش سرعت دسترسی به داده ها شود.(۴)

فرض کنید یک پایگاه داده دارای جدولی به نام "کارکنان" است که حاوی ستون "نام" است. اگر این ستون حاوی نام های تکراری باشد، می توان از فشرده سازی ساده برای کاهش حجم داده های ذخیره شده استفاده کرد. به عنوان مثال، اگر ستون "نام" حاوی موارد زیر باشد:

علی
علی
محمد
حسین
علی

با استفاده از فشرده سازی ساده، می توان این ستون را به صورت زیر فشرده کرد:

(علی ,۲)
(محمد ,۱)
(حسین ,۱)
(۱, علی)

۱,۳. فشرده سازی پیشرفته (Advanced compression):

فشرده سازی پیشرفته (Advanced compression) یک روش فشرده سازی است که از الگوریتم های پیچیده تر برای کاهش حجم داده ها استفاده می کند .این روش می تواند فضای بسیار کمتری

را نسبت به روش های ساده تر اشغال کند، اما سرعت آن نیز کندتر است.

فشرده سازی پیشرفته معمولاً برای داده هایی استفاده می شود که الگوهای تکراری زیادی دارند .به عنوان مثال، داده های متنی، داده های عددی، و داده های ترکیبی می توانند برای فشرده سازی پیشرفته مناسب باشند.

در فشرده سازی پیشرفته، از تکنیک هایی مانند موارد زیر استفاده می شود:

- حذف تکرارها :این تکنیک تکرارهای داده ها را حذف می کند.
- استفاده از کدهای کوتاهتر :این تکنیک مقادیر تکراری را با کدهای کوتاهتر جایگزین می کند.
- استفاده از الگوریتم های تبدیل :این الگوریتم ها داده ها را به فرمت های فشرده تبدیل می کنند.

فشرده سازی پیشرفته مزایای زیادی دارد، از جمله:

- کاهش هزینه های ذخیره سازی :فشرده سازی پیشرفته می تواند هزینه های ذخیره سازی داده ها را کاهش دهد.
- افزایش سرعت انتقال داده ها :فشرده سازی پیشرفته می تواند سرعت انتقال داده ها را افزایش دهد.
- بهبود عملکرد پایگاه داده :فشرده سازی پیشرفته می تواند عملکرد پایگاه داده را بهبود بخشد.

با این حال، فشرده سازی پیشرفته نیز معایبی دارد، از جمله:

- کاهش سرعت پردازش :فشرده سازی پیشرفته می تواند سرعت پردازش داده ها را کاهش دهد.
- افزایش پیچیدگی :فشرده سازی پیشرفته می تواند پیچیدگی پایگاه داده را افزایش دهد.

انتخاب نوع فشرده سازی مناسب به عوامل مختلفی بستگی دارد، از جمله نوع داده ها، حجم داده ها، و میزان دسترسی به داده ها .اگر داده های پایگاه داده دارای الگوهای تکراری زیادی هستند، فشرده سازی پیشرفته می تواند گزینه مناسبی باشد.

فرض کنید یک جدول در پایگاه داده داریم که حاوی اطلاعات مربوط به مشتریان است .این جدول شامل ستون هایی مانند نام، نام خانوادگی، آدرس، کد پستی، و شماره تلفن است.

اگر از فشرده سازی پیشرفته برای این جدول استفاده کنیم، می توانیم فضای زیادی را ذخیره کنیم .به عنوان مثال، می توانیم از الگوریتم های فشرده سازی متن برای فشرده سازی ستون های نام و نام خانوادگی استفاده کنیم .این الگوریتم ها می توانند با استفاده از تکنیک هایی مانند حذف تکرارها و استفاده از کدهای کوتاهتر برای مقادیر تکراری، فضای زیادی را ذخیره کنند.

به عنوان مثال، اگر نام "محمد" در جدول ۱۰۰ بار تکرار شود، می توانیم از یک کد کوتاهتر مانند "م۱۰۰" برای آن استفاده کنیم . این کار باعث می شود که فضای کمتری برای ذخیره نام "محمد" در جدول استفاده شود.

همچنین می توانیم از الگوریتم های فشرده سازی عددی برای فشرده سازی ستون های کد پستی و شماره تلفن استفاده کنیم . این الگوریتم ها می توانند با استفاده از تکنیک هایی مانند تبدیل اعداد به فرمت فشرده، فضای زیادی را ذخیره کنند.

به عنوان مثال، اگر کد پستی "۱۲۳۴۵۶۷۸۹" در جدول ۱۰۰ بار تکرار شود، می توانیم از یک کد کوتاهتر مانند "۱۲۳۴۵۶۷۸۹۰۰" برای آن استفاده کنیم .این کار باعث می شود که فضای کمتری برای ذخیره کد پستی "۱۲۳۴۵۶۷۸۹" در جدول استفاده شود.

با استفاده از این تکنیک ها، می توانیم فضای زیادی را برای جدول مشتریان ذخیره کنیم .به عنوان مثال، اگر این جدول ۱۰۰۰ رکورد داشته باشد، می توانیم با استفاده از فشرده سازی پیشرفته، فضای ذخیره سازی آن را تا ۵۰ درصد کاهش دهیم.(۵)

۲،۱.فشرده سازی بر اساس نوع داده

روش های مختلفی برای فشرده سازی داده های کاراکتر وجود دارد. برخی از این روش ها علاوه بر حذف تکرارها، از روش های دیگری مانند کدگذاری Huffman نیز استفاده می کنند. کدگذاری Huffman روشی برای فشرده سازی داده ها است که بر اساس احتمال وقوع مقادیر مختلف داده ها کار می کند.

مزایا

فشرده سازی داده های کاراکتر مزایای زیادی دارد، از جمله:

- کاهش هزینه های ذخیره سازی: فشرده سازی می تواند فضای ذخیره سازی مورد نیاز برای داده های کاراکتر را به میزان قابل توجهی کاهش دهد.
- افزایش سرعت انتقال داده ها: فشرده سازی می تواند سرعت انتقال داده های کاراکتر را افزایش دهد.
- بهبود عملکرد پایگاه داده: فشرده سازی می تواند عملکرد پایگاه داده را در برخی موارد بهبود بخشد.

معایب

فشرده سازی داده های کاراکتر همچنین معایبی دارد، از جمله:

- افزایش هزینه پردازش: فشرده سازی می تواند هزینه پردازش داده های کاراکتر را افزایش دهد.
- کاهش امنیت: فشرده سازی می تواند امنیت داده های کارا cterرا کاهش دهد.

- انتخاب نوع فشرده سازی مناسب برای یک پایگاه داده به عوامل مختلفی بستگی دارد، از جمله نوع داده ها، حجم داده ها، و میزان دسترسی به داده ها.

فرض کنید بخواهیم جمله "این جمله یک مثال از فشرده سازی داده های کاراکتر است" را فشرده کنیم.

در این جمله، کاراکترهای "ا"، "ی"، "ن"، و "ه" تکرار شده اند. می توان از این تکرارها برای فشرده سازی جمله استفاده کرد.

برای مثال، می توان از کدهای زیر برای فشرده سازی جمله استفاده کرد:

i = 1
n = 2
h = 3

با استفاده از این کدها، جمله "این جمله یک مثال از فشرده سازی داده های کاراکتر است" می تواند به صورت زیر فشرده شود:

i1n2h3e1m2p2l2s2a1d1d2a1t2i1s2e1n2

این جمله فشرده شده، ۲۹ کاراکتر دارد، در حالی که جمله اصلی ۵۵ کاراکتر دارد. بنابراین، با استفاده از فشرده سازی داده های کاراکتر، می توان حجم جمله را ۴۵ درصد کاهش داد.

فرض کنید بخواهیم متن زیر را فشرده کنیم:

این یک متن آزمایشی است که برای نشان دادن مزایای فشرده سازی داده های کاراکتر استفاده می شود. در این متن، از کلمات و عبارات تکراری زیادی استفاده شده است.

در این متن، کلمات و عبارات "این"، "متن"، "آزمایشی"، "است"، "برای"، "نشان دادن"، "مزایای"، "فشرده سازی"، "داده های"، "کاراکتر"، و "استفاده" تکرار شده اند. می توان از این تکرارها برای فشرده سازی متن استفاده کرد.

برای مثال، می توان از کدهای زیر برای فشرده سازی متن استفاده کرد:

i = 1
t = 2
e = 3
m = 4
a = 5
s = 6

با استفاده از این کدها، متن "این یک متن آزمایشی است که برای نشان دادن مزایای فشرده سازی داده های کاراکتر استفاده می شود" می تواند به صورت زیر فشرده شود:

i1t2e3m4a5s6e1i1n1t2e1s1t2a1d1i1l1s1i1u1z1a1n1i1
e1s1f1i1l1t1i1z1a1t1i1o1n1s1

این متن فشرده شده، ۹۴ کاراکتر دارد، در حالی که متن اصلی ۱۹۸ کاراکتر دارد. بنابراین، با استفاده از فشرده سازی داده های کاراکتر، می توان حجم متن را ۵۲ درصد کاهش داد.(۳)

۲,۲.فشرده سازی بر اساس نوع داده

فشرده سازی داده های عددی روشی برای کاهش حجم داده های عددی ذخیره شده در پایگاه داده است. این کار با استفاده از الگوریتم های مختلف انجام می شود.

یکی از روش های فشرده سازی داده های عددی، استفاده از کدهای کوتاهتر برای مقادیر تکراری است. به عنوان مثال، اگر در یک جدول، سن اکثر افراد بین ۲۵ تا ۲۷ سال باشد، می توان سن ۲۵ را با کد ۱، سن ۲۶ را با کد ۲، و سن ۲۷ را با کد ۳ کدگذاری کرد. در این صورت، حجم جدول فشرده شده به نصف حجم جدول اصلی کاهش می یابد.

روش دیگر فشرده سازی داده های عددی، استفاده از الگوریتم های فشرده سازی خاص است. این الگوریتم ها معمولاً پیچیده تر از روش اول هستند، اما می توانند کارایی بیشتری داشته باشند.

فشرده سازی داده های عددی مزایای زیادی دارد، از جمله:

- کاهش هزینه های ذخیره سازی
- افزایش سرعت انتقال داده ها
- بهبود عملکرد پایگاه داده
- انتخاب نوع فشرده سازی مناسب برای داده های عددی به عوامل مختلفی بستگی دارد، از جمله نوع داده ها، حجم داده ها، و میزان دسترسی به داده ها.

در ادامه، به برخی از روش های فشرده سازی داده های عددی اشاره می کنیم:

فشرده سازی با استفاده از کدهای کوتاهتر: این روش ساده ترین روش فشرده سازی داده های عددی است. در این روش، مقادیر تکراری با کدهای کوتاهتر کدگذاری می شوند.

فشرده سازی با استفاده از الگوریتم های خاص: این روش ها پیچیده تر از روش اول هستند، اما می توانند کارایی بیشتری داشته باشند. برخی از الگوریتم های فشرده سازی خاص برای داده های عددی عبارتند از:

- الگوریتم فشرده سازیHuffman
- الگوریتم فشرده سازیLZW
- الگوریتم فشرده سازیRun-length encoding

فشرده سازی داده های عددی یکی از تکنیک های مهم در پایگاه داده ها است. این تکنیک می تواند با کاهش حجم داده ها، مزایای زیادی برای پایگاه داده ها به همراه داشته باشد.(۸)

فرض کنید یک جدول داریم که در آن سن افراد ذخیره شده است. سن افراد در این جدول به صورت زیر است:

سن
۲۵
۲۵
۲۶
۲۶
۲۷

در این جدول، سن اکثر افراد بین ۲۵ تا ۲۷ سال است. بنابراین، می توان از فشرده سازی داده های عددی برای کاهش حجم این جدول استفاده کرد.

یک روش ساده برای فشرده سازی این جدول، استفاده از کدهای کوتاهتر برای مقادیر تکراری است. به عنوان مثال، می توان سن ۲۵ را با کد ۱، سن ۲۶ را با کد ۲، و سن ۲۷ را با کد ۳ کدگذاری کرد. در این صورت، جدول فشرده شده به صورت زیر خواهد بود:

کد
۱
۱
۲
۲
۳

با استفاده از این روش، حجم جدول فشرده شده به نصف حجم جدول اصلی کاهش می یابد.

۲,۳. فشرده سازی داده های ترکیبی (Mixed data compression)

فشرده سازی داده های ترکیبی (Mixed data compression) روشی برای فشرده سازی داده هایی است که از انواع مختلف داده ها تشکیل شده اند. این روش معمولاً ترکیبی از روش های مختلف فشرده سازی را برای بهینه سازی فضای ذخیره سازی استفاده می کند.

یکی از مزایای فشرده سازی داده های ترکیبی این است که می تواند فضای ذخیره سازی زیادی را صرفه جویی کند. این امر می تواند هزینه های ذخیره سازی را کاهش دهد و عملکرد پایگاه داده را بهبود بخشد.

برای فشرده سازی داده های ترکیبی، ابتدا باید داده ها را به اجزای تشکیل دهنده آنها تقسیم کرد. سپس، می توان از روش های مختلف فشرده سازی برای فشرده سازی هر یک از اجزای داده استفاده کرد.

در مثال پایین، ما داده های پایگاه داده مشتریان را به سه جزء نام، نام خانوادگی، و آدرس تقسیم کردیم. سپس، از روش فشرده سازی کاراکتر برای فشرده سازی نام و نام خانوادگی، از روش فشرده سازی متنی برای فشرده سازی آدرس، و از روش فشرده سازی عددی برای فشرده سازی شماره تلفن استفاده کردیم.البته، انتخاب روش های فشرده سازی مناسب برای هر یک از اجزای داده به عوامل مختلفی بستگی دارد، از جمله نوع داده، میزان تکرار، و میزان دسترسی به داده.

مثال: فرض کنید یک پایگاه داده مشتریان داریم که شامل نام، نام خانوادگی، آدرس، و شماره تلفن مشتریان است. نام و نام خانوادگی مشتریان معمولاً از نوع داده کاراکتر هستند، آدرس معمولاً از نوع داده متنی است، و شماره تلفن معمولاً از نوع داده عددی است.برای فشرده سازی داده های این پایگاه داده می توانیم از روش فشرده سازی داده های ترکیبی استفاده کنیم. در این روش، از ترکیبی از روش های مختلف فشرده سازی برای داده های مختلف استفاده می شود.برای فشرده سازی نام و نام خانوادگی مشتریان می توانیم از روش فشرده سازی کاراکتر استفاده کنیم. این روش می تواند با استفاده از الگوریتم های مختلف، فضای زیادی را برای داده های کاراکتر ذخیره کند.برای فشرده سازی آدرس مشتریان می توانیم از روش فشرده سازی متنی استفاده کنیم. این روش می تواند با استفاده از الگوریتم های مختلف، فضای زیادی را برای داده های متنی ذخیره کند.برای فشرده سازی شماره تلفن مشتریان می توانیم از روش فشرده سازی عددی استفاده کنیم. این روش می تواند با استفاده از روش های مختلف، فضای زیادی را برای داده های عددی ذخیره کند. (۲)

۳,۱. فشرده سازی در سطح جدول (Table-level compression)

فشرده سازی در سطح جدول روشی برای فشرده سازی کل جدول است. در این روش، کل داده های جدول فشرده می شوند و یک نسخه فشرده از جدول ایجاد می شود.

فشرده سازی در سطح جدول مزایای زیادی دارد، از جمله:

- کاهش حجم داده های ذخیره شده
- افزایش سرعت انتقال داده ها
- بهبود عملکرد پایگاه داده

فشرده سازی در سطح جدول ساده ترین روش فشرده سازی پایگاه داده است. این روش انعطاف پذیری کمتری نسبت به فشرده سازی در سطح رکورد دارد، اما سرعت اجرای آن نیز بیشتر است.

فشرده سازی در سطح جدول معمولاً برای جدول هایی که دارای داده های تکراری هستند استفاده می شود. به عنوان مثال، می توان از این روش برای فشرده سازی جدول های مشتریان، محصولات، و سفارشات استفاده کرد.

در مثال پایین، فشرده سازی در سطح جدول باعث کاهش حجم داده های جدول مشتریان از ۱۲۰ بایت به ۴۸ بایت شد. این کاهش حجم باعث افزایش سرعت انتقال داده ها و بهبود عملکرد پایگاه داده می شود.

فرض کنید یک جدول با نام مشتریان داریم که شامل اطلاعات زیر است:

نام	آدرس	تلفن
علی	تهران	۰۹۱۲۳۴۵۶۷۸۹
محمد	اصفهان	۰۹۱۲۱۲۳۴۵۶۷
حسین	مشهد	۰۹۱۲۲۲۳۳۴۴۵

در این جدول، نام و آدرس مشتریان در ستون های نام و آدرس تکرار شده است. با استفاده از فشرده سازی در سطح جدول، می توان این تکرارها را حذف کرد و فضای ذخیره سازی را کاهش داد.

به عنوان مثال، می توان از الگوریتم فشرده سازی (Run-Length Encoding) RLE برای فشرده سازی ستون نام استفاده کرد. این الگوریتم تکرارهای متوالی را با یک کد کوتاهتر جایگزین می کند.

در مثال بالا، می توان ستون نام را به صورت زیر فشرده کرد:

نام	آدرس	تلفن
علی	تهران	۰۹۱۲۳۴۵۶۷۸۹
محمد	اصفهان	۰۹۱۲۱۲۳۴۵۶۷
حسین	مشهد	۰۹۱۲۲۲۳۳۴۴۵

در این حالت، فضای ذخیره سازی ستون نام از ۳۰ بایت به ۱۲ بایت کاهش یافته است.(۶)

۳,۲. فشرده سازی در سطح رکورد (Row-level compression)

فشرده سازی در سطح رکورد انعطاف پذیری بیشتری نسبت به فشرده سازی در سطح جدول دارد. این بدان معناست که می توان داده های هر رکورد را به طور مستقل فشرده کرد. این امر می تواند به ویژه برای داده هایی که مقادیر تکراری زیادی دارند مفید باشد.

با این حال، فشرده سازی در سطح رکورد می تواند پیچیده تر از فشرده سازی در سطح جدول باشد. این امر به این دلیل است که باید الگوریتم فشرده سازی برای هر نوع داده ای که در جدول وجود دارد، پیاده سازی شود.

مزایا

- انعطاف پذیری بیشتر
- کاهش بیشتر حجم داده ها

معایب

- پیچیدگی بیشتر
- کاهش عملکرد در برخی موارد

مثال: فرض کنید یک جدول با نام کارمندان داریم که دارای ستون های زیر است:

ستون	نوع داده
کد کارمند	عدد صحیح
نام	رشته
سن	عدد صحیح
جنسیت	متن

اگر از فشرده سازی در سطح رکورد استفاده کنیم، هر رکورد جداگانه فشرده می شود. به عنوان مثال، رکورد زیر:

۱۲۳",علیرضا محمدی",۳۰,"مرد"

می تواند به صورت زیر فشرده شود:

۱۲۳",علیرضا محمدی",۳۰,"م"

در این مثال، جنسیت "مرد" به صورت "م" فشرده شده است. این کار باعث کاهش حجم رکورد از ۱۶ بایت به ۱۲ بایت می شود.(۴)

۳,۳. فشرده سازی در سطح صفحه (Page-level compression)

فشرده سازی در سطح صفحه (Page-level compression) روشی است که در آن داده های یک صفحه فشرده می شوند. یک صفحه در پایگاه داده معمولاً ۸ کیلوبایت یا ۱۶ کیلوبایت است.

فشرده سازی در سطح صفحه مزایای زیادی دارد، از جمله:

- کاهش حجم داده های ذخیره شده در پایگاه داده
- افزایش سرعت انتقال داده ها
- بهبود عملکرد پایگاه داده

فشرده سازی در سطح صفحه می تواند با استفاده از روش های مختلفی انجام شود. برخی از این روش ها عبارتند از:

- فشرده سازی رکورد :(Row compression) در این روش، داده های هر رکورد فشرده می شوند. برای مثال، اگر نام یک رکورد "علی" باشد، می توان آن را به صورت "علی" فشرده کرد.
- فشرده سازی ستون :(Column compression) در این روش، داده های یک ستون فشرده می شوند. برای مثال، اگر ستون سن در جدول فقط مقادیر ۱۸، ۲۰، ۲۵، و ۳۰ را داشته باشد، می توان آن را به صورت "۱۸،۲۰،۲۵،۳۰" فشرده کرد.
- فشرده سازی دیکشنری (Dictionary compression):در این روش، یک دیکشنری از مقادیر تکراری ایجاد می شود و به جای مقادیر تکراری، از کدهای کوتاهتر استفاده می شود. برای مثال، اگر ستون نام در جدول فقط مقادیر "علی"، "محمد"، "رضا"، و "حسین" را داشته باشد، می توان یک دیکشنری از این مقادیر ایجاد کرد و به جای مقادیر تکراری، از کدهای کوتاهتری مانند "۱"، "۲"، "۳"، و "۴" استفاده کرد.

انتخاب روش فشرده سازی مناسب به عوامل مختلفی بستگی دارد، از جمله نوع داده ها، حجم داده ها، و میزان دسترسی به داده ها.
فشرده سازی در سطح صفحه، برای پایگاه داده هایی که حجم زیادی از داده های تکراری دارند، مناسب است. به عنوان مثال، پایگاه داده های CRM (مدیریت ارتباط با مشتری)، پایگاه داده های ERP (برنامه ریزی منابع سازمانی)، و پایگاه داده های مالی، از جمله پایگاه داده هایی هستند که برای فشرده سازی در سطح صفحه مناسب هستند.

مثال:

فرض کنید یک پایگاه داده دارای جدولی با نام students است که شامل ستون های زیر است:

id: یک عدد صحیح

name: یک رشته

age: یک عدد صحیح

اگر این جدول ۱۰۰۰ رکورد داشته باشد، حجم آن در حالت عادی برابر با ۱۰۰۰ * ۴ بایت = ۴۰۰۰۰ بایت خواهد بود.
با استفاده از فشرده سازی در سطح صفحه، می توان حجم این جدول را به میزان قابل توجهی کاهش داد. به عنوان مثال، با استفاده از فشرده سازی داده های کاراکتر، می توان ستون name را با استفاده از کدگذاری Huffman فشرده کرد. این کدگذاری، حروف و عبارات تکراری را با کدهای کوتاهتر جایگزین می کند.

با استفاده از این کدگذاری، حجم ستون name به میزان قابل توجهی کاهش می یابد. به عنوان مثال، اگر ۲۰٪ از رکوردهای جدول دارای نام تکراری باشند، می توان حجم این ستون را به ۶۰٪ کاهش داد. در این صورت، حجم کل جدول به ۴۰۰۰۰ * ۰,۶ = ۲۴۰۰۰ بایت کاهش می یابد.
در پایگاه داده هایSQL Server ، می توان از فشرده سازی در سطح صفحه برای فشرده سازی جداول، اشیا، و کل پایگاه داده استفاده کرد. برای فشرده سازی یک جدول درSQL Server ، می توان از دستور ALTER TABLE استفاده کرد. به عنوان مثال، برای فشرده سازی جدول students در مثال قبل، می توان از دستور زیر استفاده کرد:

```
ALTER TABLE students
WITH (DATA_COMPRESSION = PAGE);
```

این دستور باعث می شود که داده های جدول students در سطح صفحه فشرده شوند.(۵)

انواع الگوریتم های فشرده سازی

الگوریتم فشرده سازی هافمن(Huffman)

الگوریتم فشرده سازی هافمن یک الگوریتم فشرده سازی بدون ضرر است که بر اساس استفاده از کدهای پیشوندی برای نمادها یا کاراکترها کار می کند. این الگوریتم، نمادهای تکراری را با کدهای کوتاهتر و نمادهای غیرتکرار را با کدهای طولانی‌تر جایگزین می‌کند.
الگوریتم فشرده سازی هافمن به صورت زیر کار می کند:

- ابتدا، نمادها یا کاراکترها بر اساس احتمال وقوعشان مرتب می شوند.
- سپس، نمادها یا کاراکترهای با احتمال وقوع کمتر به صورت جفت گروه بندی می شوند.
- برای هر جفت، کد پیشوندی اختصاص داده می شود که از ترکیب کدهای پیشوندی نمادهای تشکیل دهنده آن جفت تشکیل شده است.
- این فرآیند تکرار می شود تا زمانی که فقط یک نماد یا کاراکتر باقی بماند.

الگوریتم فشرده سازی هافمن یک الگوریتم فشرده سازی کارآمد است که می تواند برای فشرده سازی انواع مختلف داده ها استفاده شود(۸).

الگوریتم فشرده سازی(LZW (Lempel-Ziv-Welch

الگوریتم فشرده سازی LZW یک الگوریتم فشرده سازی بدون ضرر است که بر اساس استفاده از یک واژه نامه برای ذخیره نمادها یا کاراکترهای تکراری کار می کند. این الگوریتم، نمادهای تکراری را با یک نماد جدید جایگزین می کند که در واژه نامه ذخیره شده است.

الگوریتم فشرده سازی LZW به صورت زیر کار می کند:

- ابتدا، یک واژه نامه خالی ایجاد می شود.
- سپس، اولین نماد ورودی به واژه نامه اضافه می شود.
- برای هر نماد بعدی ورودی، اگر آن نماد در واژه نامه موجود باشد، آن نماد از واژه نامه خوانده می شود.
- اگر آن نماد در واژه نامه موجود نباشد، آن نماد با یک نماد جدید جایگزین می شود که از ترکیب نماد قبلی و نماد جدید تشکیل شده است.
- این فرآیند تکرار می شود تا زمانی که همه نمادهای ورودی پردازش شوند.

الگوریتم فشرده سازی LZW یک الگوریتم فشرده سازی کارآمد است که می تواند برای فشرده سازی انواع مختلف داده ها استفاده شود.(۸)

الگوریتم فشرده سازی(Run-length encoding (RLE

الگوریتم فشرده سازی Run-length encoding یک الگوریتم فشرده سازی بدون ضرر است که بر اساس استفاده از کدهای پیشوندی برای نمادهای یا کاراکترهای تکراری کار می کند. این الگوریتم، نمادهای تکراری را با یک کد پیشوندی که شامل تعداد تکرارها و نماد تکراری است، جایگزین می کند.

الگوریتم فشرده سازی Run-length encoding به صورت زیر کار می کند:

ابتدا، یک متغیر count برای ذخیره تعداد تکرارها مقداردهی اولیه می شود.

سپس، اولین نماد ورودی خوانده می شود.

اگر نماد فعلی با نماد قبلی برابر باشد، مقدار count یک واحد افزایش می یابد.

اگر نماد فعلی با نماد قبلی برابر نباشد، کد پیشوندی شامل تعداد تکرارها و نماد تکراری به خروجی نوشته می شود.

مقدار count مقداردهی اولیه می شود.

این فرآیند تکرار می شود تا زمانی که همه نمادهای ورودی پردازش شوند.

الگوریتم فشرده سازی Run-length encoding یک الگوریتم فشرده سازی ساده است که می تواند برای فشرده سازی داده هایی که شامل مقادیر تکراری هستند، استفاده شود.

در ادامه، مقایسه ای بین این سه الگوریتم فشرده سازی ارائه شده است:

ویژگی	الگوریتم هافمن	الگوریتم LZW	الگوریتم Run-length encoding
نوع	بدون ضرر	بدون ضرر	بدون ضرر
پیچیدگی	متوسط	متوسط	ساده
کارایی	بالا	بالا	متوسط
کاربرد	انواع داده ها	انواع داده ها	داده هایی که شامل مقادیر تکراری هستند

الگوریتم هافمن و LZW الگوریتم های فشرده سازی کارآمدی هستند که می توانند برای فشرده سازی انواع مختلف داده ها استفاده شوند. الگوریتم Run-length encoding یک الگوریتم فشرده سازی ساده است که می تواند برای فشرده سازی داده هایی که شامل مقادیر تکراری هستند، استفاده شود.(۹)

نتیجه‌گیری

فشرده سازی پایگاه داده روشی برای کاهش حجم داده های ذخیره شده در پایگاه داده است. این کار با حذف اطلاعات اضافی و استفاده از کدهای کوتاه تر برای مقادیر تکراری انجام می شود. فشرده سازی پایگاه داده مزایای زیادی دارد، از جمله:

- کاهش هزینه های ذخیره سازی
- افزایش سرعت انتقال داده ها
- بهبود عملکرد پایگاه داده

انواع مختلفی از فشرده سازی پایگاه داده وجود دارد که بر اساس سطح فشرده سازی، نوع داده، و مکان فشرده سازی تقسیم بندی می شوند. انتخاب نوع فشرده سازی مناسب برای یک پایگاه داده به عوامل مختلفی بستگی دارد، از جمله نوع داده ها، حجم داده ها، و میزان دسترسی به داده ها.

در این مقاله، انواع مختلف فشرده سازی پایگاه داده و الگوریتم های فشرده سازی رایج مورد بررسی قرار گرفت. همچنین، سه مقاله برای توضیحات الگوریتمی فشرده سازی معرفی شد.

در نتیجه، فشرده سازی پایگاه داده یک روش مهم برای بهبود عملکرد و کارایی پایگاه داده ها است.(۶)

مراجع

۱. Aggarwal, Charu C., and ChengXiang Zhai. "Mining text data." Springer, 2012) .Aggarwal و Zhai ، (۲۰۱۲

۲. Keogh, Eamonn, et al. "The UCR Time Series Classification/Clustering Homepage." University of California, Riverside) .Keogh(و همکاران، بی‌تاریخ

۳. Chaudhuri, Surajit, and Umeshwar Dayal. "An overview of data warehousing and OLAP technology." ACM Sigmod Record 26.1 (1997): 65-74) .Chaudhuri و Dayal(۱۹۹۷ ،

۴. A. Kemal and B. Kalyanaraman, "Data Compression Techniques for Database Systems," in Proceedings of the IEEE, vol. 100, no. 9, pp. 1-15, (منبع: ۲۰۱۲ .A. Kemal و B. Kalyanaraman(

۵. H. Kim, J. Lee, and C. Park, "A Survey on Compression Techniques in Database Management Systems," Journal of Information Science and Engineering, vol. 30, no. 6, pp. 2123-2145, 2014:منبع) . H. Kim ،J. Lee و C. Park(

۶. S. Acharya and D. B. Gibbons, "Improving Data Management with Compression in Database Systems," ACM SIGMOD Record, vol. 29, no. 3, pp. ۳۱۵-۳۲۷, ۲۰۰۰ :منبع) .S. Acharya و D. B. Gibbons(

۷. P. A. Boncz, M. Zukowski, and N. Nes, "MonetDB/X100: Hyper-Pipelining Query Execution," in Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, pp. 550-561, 2005 :منبع) .P. A. Boncz ،M. Zukowski و N. Nes(

۸. Cover, Thomas M., and Joy A. Thomas. "Elements of information theory." Wiley, 2006. (Cover و Thomas۲۰۰۶ ،)

۹. Aggarwal, Charu C., and ChengXiang Zhai. "Mining text data." Springer, 2012. (Aggarwal و Zhai ، ۲۰۱۲)